Efficient Approximation Algorithms for Optimal Large-scale Network Monitoring

Michael G. Kallitsis, Stilian Stoev and George Michailidis
Department of Statistics, University of Michigan, Ann Arbor
mgkallit@umich.edu, sstoev@umich.edu, gmichail@umich.edu

***Abstract*—The growing amount of applications that generate vast amount of data in short time scales render the problem of *partial monitoring*, coupled with prediction, a rather fundamental one. We study the aforementioned canonical problem under the context of *large-scale* monitoring of communication networks. We consider the problem of *selecting* the "best" subset of links so as to *optimally predict* the quantity of interest at the remaining ones. This is a well know NP-hard problem, and algorithms seeking the exact solution are prohibitively expensive. We present a number of *approximation algorithms* that: 1) their computational complexity gains a significant improvement over existing greedy algorithms; 2) exploit the geometry of *principal component analysis*, which also helps us establish theoretical bounds on the prediction error; 3) are amenable for randomized implementation and execution in parallel or distributed fashion, a process that often yields the exact solution. The new algorithms are demonstrated and evaluated using real-world network data.**

I. Introduction

A. Motivation

ADVANCES in high-throughput technologies for information collection have led to unprecedented growth of data that need to be analyzed and interpreted. Resource constraints, however, impose limitations on the amount of data that can be processed. For example, in network monitoring, limited bandwidth prevents all data from being sent to a centralized coordinator (*e.g.*, Cisco's Netflow Collector [1]). In wireless sensor applications, such as environmental monitoring, sensors' power constraints dictate a wise utilization of the available resources. Therefore, designing efficient algorithms for information learning through prediction is an important problem that needs to be tackled. This modeling approach allows for *online* prediction of the information of interest, by utilizing measurements of a much smaller set of variables. Obviously, the accuracy of the prediction heavily depends on the selection of variables to be monitored. One aims to minimize the *uncertainty* of the information learned. This uncertainty is usually expressed via the covariance matrix of the prediction error. This is a canonical *design* problem with various applications such as, optimal monitoring of computer networks for identification of traffic anomalies [2], stock market prediction [3], sensors' placement for environmental monitoring (see [4] and references therein), transportation network design [5], etc. In this study, we develop fast algorithms addressing the problem at hand in the context of network monitoring; namely, how to select K network links, so as to "best" predict link utilization on the remaining ones (see Figures 8 and 9).

Online monitoring for detection and classification of anomalies on a computer network's traffic is critical for its smooth operation, especially in the presence of new protocols (*e.g.*, peer-to-peer) and applications such as social networking and cloud computing. In [2], the authors propose a method based on *principal component analysis* (PCA) for anomaly detection by classifying traffic measurements into the categories of normal or anomalous. In [6], a scheme for fault detection based on the distributional characteristics of traffic is introduced that is able to capture spatial and temporal abnormal activities. In both studies, periodic traffic flow measurements must be collected on a *large set* of links, a costly and computationally challenging task. In [7], this problem is partly alleviated by considering measuring aggregate traffic coupled with a sampling mechanism that significantly reduces the data collection and processing overhead.

Global monitoring of large-scale networks, however, involves large traffic volumes, and becomes impractical and often impossible due to resource constraints. Nevertheless, using recent work on global traffic modeling [8], *network prediction* and *kriging* [9], [10] one can develop an accurate statistical model of the evolution of traffic flows across an entire network by monitoring only a *subset* of network links. Namely, by taking into account the routing of traffic flows and the long-range dependence in time of traffic traces, one obtains a statistical model for the dependence between the traffic loads on all links in the network. This model can then be exploited for optimal traffic prediction.

To informally introduce the problem under study, consider the toy example of Fig. 1 based on the Internet2 network [11]. The active links consist of the highlighted

This research was partially supported by the NSF grant DMS-0806094 at the University of Michigan, Ann Arbor.

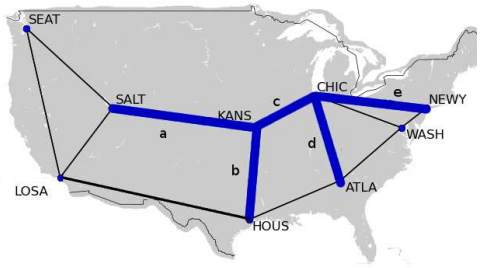


Fig. 1. A toy example on a subnetwork of the Internet2 topology. For the real-world application on the full topology see Section VII.

ones. Suppose we have the following 5 network flows with the same statistical characteristics, symbolized as (S, D) pairs where S is the source node and D the destination: (SALT, NEWY), (SALT, ATLA), (HOUS, NEWY), (HOUS, ATLA) and (KANS, ATLA). Assuming that we have specified our favorite criterion for measuring the prediction uncertainty, we ask the following question: Should we had to choose only two links so as to optimally predict the traffic at the remaining ones what would the chosen set be? One can easily guess that the first member of that set should be link c. This is because link c serves the most flows; indeed, link c belongs to the optimal set (in general, more than one might exist). The other link can be either d or e. Finally, the less obvious choice $\{d, e\}$ is also an optimal solution.

The situation becomes trickier if we assign link preferences. For example, a network operator could face different scenarios and may assign distinct link weights/priorities, such as monitoring links which can be more error prone, or links that are more susceptible to network anomalies, or even not wanting to observe inexpensive links, etc. In our previous network topology, link a is inserted into the observed set – together with link c – when its priority elevates. To make matters more complicated, assume that flows are correlated and have different traffic characteristics. Moreover, add the requirement that these decisions should be made online and fast, since network conditions and/or operators' preferences change frequently. Amid all these, add the fact that in real-life we have to deal with hundreds or thousands of links and traffic flows. One can then see a complete picture of how hard this large-scale monitoring problem can become, and that efficient approximation algorithms could be very appealing to network managers.

B. Contributions

We study the fundamental problem of finding the subset of links to monitor which yields best overall prediction for the remaining ones. The objective is, given a "budget" of K links to be selected (out of a total of L), to determine the optimal set of them in order to predict the network traffic on the unobserved ones with minimum uncertainty. We formulate a canonical model (see Section III-A) which we then tailor for tackling the challenging problem of large-scale network monitoring. We emphasize that our algorithms can be readily applied to any other optimal learning problem that fits under the framework of Section III-A. In the canonical problem, we aim to minimize a functional of the prediction error, which represents the uncertainty on the information that is learned. There are several popular criteria for measuring the uncertainty, such as entropy (also known as D-optimality, see [12]) of the design matrix, mutual information [4], trace (A-optimality) and spectral norm (E-optimality) [13]. In this work, we measure the quality of the prediction by investigating the criteria of A- and E-optimality.

Optimal prediction belongs to the category of *subset selection* problems, which are combinatorial, NP-hard problems. For example, min- and max-entropy (D-optimality) sampling are shown to be NP-hard problems in [12] by reduction from the well-known NP-hard problems of CLIQUE and STABLE SET [14]. NP-hardness of our problem remains in open problem. However, a proof of NP-hardness for an analogous problem is given in the Appendix. In principle, exact solutions can be obtained by using an *integer program* formulation. However, the problem becomes prohibitively expensive for even moderate-size datasets. For example, an exhaustive search implementation in CPLEX running on a medium-sized computing cluster required several hours to complete the solution for a 26 variables problem with a budget of 12. Other methods, such as the mixed-integer program with constraint generation proposed in [15], are also computationally expensive and impractical. The latter algorithm requires also *submodularity* for the objective set function. Unfortunately, submodularity, which is similar to the convexity property for real valued functions, is not satisfied for all the objective criteria of interest. Branch and bound methods could be applied (see [12]), but the huge number of "branching" choices makes the procedure unattractive for online implementations in large-scale networks. These challenges emphasize the need for approximate, but *fast* heuristic algorithms for solving the optimal prediction problem.

The main contributions of our study are:

1) We introduce new heuristic algorithms that exhibit superior computational complexity to existing ones. The complexity of our algorithms ranges from $O(Kn^2)$ to $O(Kn^3)$, a significant improvement over existing greedy algorithms with $O(Kn^4)$ complexity, where n represents the total number of variables. These efficiency gains allow one to solve the optimal design problem in a *dynamic*, *online* fashion.
2) By exploiting the geometry of the objective functions used, together with connections to PCA we obtain *prediction error bounds* that do not require submodularity of the objective functions as in exist-

ing work (*e.g.* see [4]). Furthermore, these bounds help us assess the quality of our approximate solution with respect to the optimal solution.

3) Randomized versions of the proposed algorithms can be implemented in a parallel or distributed fashion. This leads to further improvements in practice, often yielding the optimal solution (see Fig. 3(a)).

The remainder of the paper is organized as follows: in Section II we outline the related work. In Section III we formally define the canonical problem, followed by the network monitoring problem. In Section IV we discuss the connections with PCA and derive our PCA-based lower bound on the performance of our algorithms. Section V introduces and analyzes our three approximation algorithms, and concludes with possible extensions to their randomized versions. Section VI discusses theoretical performance guarantees for the error reduction achieved at each step. Next, in Section VII, we evaluate our methods in a variety of network scenarios, including the real-world dataset obtained from the Internet2 topology [11]. We also explain how a network operator can choose the budget K, and how weighted link monitoring can be achieved.

II. Literature Survey

The combinatorial optimization problem at hand – formally introduced in Section III-A, see Eq. (6) – belongs to the family of *subset selection* problems, and is encountered in scientific areas ranging from computer science and engineering to statistics and linear algebra. For example, in [15], a bank location problem is formulated as an integer program and algorithms based on branch and bound techniques are employed to obtain the exact solution. In [16], approximation algorithms are studied that exploit the submodularity property of the objective function. Polynomial-time, greedy heuristics are offered that yield a solution within $(1 - 1/e)$ of the optimum, where e is the base of the natural logarithm.

In [4], near-optimal sensor placement for temperature monitoring is examined. Temperatures at different locations are modeled with a multivariate Gaussian joint distribution. The objective is to place a given small number of sensors so as to optimally predict the temperature at other locations. The aim is to maximize the *mutual information* criterion and, thus, the problem is a variant of (6). The authors exploit submodularity and propose heuristics that outperform the classical implementation of the greedy algorithm. However, these heuristics are faster than classical greedy only in special cases such as when the covariance matrix of the joint Gaussian distribution has "low-bandwidth"[1]. The structure of the covariance matrix is also exploited in [17] where the problem of *subset selection for regression* is studied. This is similar in spirit to our problem under the trace criterion. The authors propose fast, exact algorithms based on dynamic programming, which apply, however, only to the special cases of "low-bandwidth" and "tree covariance" graphs. Unfortunately, such special cases do not commonly arise in real-world applications. For example, in the network monitoring problem, the topology and routing of real-life networks often lead to covariance matrices with complex structure. This motivated us to seek alternatives to the algorithms in [4], [17], which can be applied to large-scale real-world applications.

In mathematics and signal processing community the, so called, *sparse approximation* problem [18], [19], [20] is also similar to (6). In that context, the goal is to find a "sparse" subset of a dictionary $\mathcal{D} = \{\boldsymbol{\phi}_i\}_{i\in\{1,\ldots,|\mathcal{D}|\}}$ of unit vectors that span $\mathbb{R}^N$, so that a linear combination of the selected vectors best approximates a given signal $A \in \mathbb{R}^N$. Two common greedy approaches, namely *matching-pursuit* [18] and *orthogonal matching pursuit* [21], are discussed in [19] and a two-step greedy heuristic with performance guarantees is proposed.

Subset selection also appears in *variable selection* problems [22], [3], [23], that seek the best subset of features of a given design matrix. The problem has been extensively studied by the linear algebra and statistics community. In [3], a randomized procedure followed by a deterministic one is proposed, such that the K "best" variables that are returned by this two-phase algorithm capture the same amount of information as does the subspace spanned by the top K eigenvectors of the data matrix. In [23], one commonly used technique, namely the *Lasso method*, relaxes the "budgetary" constraint on the number of variables to be selected by using an alternate one on the regression coefficients vector. By doing so, the integrality constraint vanishes, and the problem can then be solved using standard quadratic programming algorithms.

Recent work on combinatorial design problems that minimize the uncertainty of the information of interest appear in [24], [25]. In [24], an approximation algorithm based on semidefinite relaxations is proposed to solve the problem of finding the least squares solution $\boldsymbol{s}$ of the system of equations $H\boldsymbol{s} = \boldsymbol{y}$, where $\boldsymbol{s}$ is a vector of binary variables and H is a matrix with bounded uncertainty. In [25], the "best" subset of s out of n wireless sensors is sought, so as to maximize the detection performance of the observed phenomena.

III. Problem Formulation

A. A Canonical Framework

We focus on instantaneous prediction, known as *kriging*. In this case, one can capture the information on a set of *observed* variables $\boldsymbol{y}_o(t) = (y_\ell(t))_{\ell\in\mathcal{O}}$, $\mathcal{O} \subseteq \mathcal{L} := \{1, \cdots, L\}$ (*e.g.* see Figures 8 and 9). The goal is

[1]A covariance matrix Σ has bandwidth β when the variables can be ordered in such a way that $\Sigma_{ij} = 0$ when $|j - i| > \beta$.

then to predict the information carried on the unobserved variables $\boldsymbol{y}_u(t) = (y_\ell(t))_{\ell\in\mathcal{U}}$, $\mathcal{U} := \{1,\cdots,L\}\setminus\mathcal{O}$, at the same time t. (For an example of temporal prediction, see [9]; henceforth, we often omit the argument t.) For jointly Gaussian random variables, the *ordinary kriging* estimate

$$\widehat{\boldsymbol{y}}_u = \boldsymbol{\mu}_u + \Sigma_{uo}\Sigma_{oo}^{-1}(\boldsymbol{y}_o - \boldsymbol{\mu}_o), \tag{1}$$

is the best linear unbiased predictor (BLUP) for $\boldsymbol{y}_u$ via $\boldsymbol{y}_o$, where $\boldsymbol{\mu}_{\mathbf{y}} = \begin{pmatrix} \boldsymbol{\mu}_u \\ \boldsymbol{\mu}_o \end{pmatrix}$ and $\Sigma_{\mathbf{y}} = \begin{pmatrix} \Sigma_{uu} & \Sigma_{uo} \\ \Sigma_{ou} & \Sigma_{oo} \end{pmatrix}$ are the partitions of the mean and the covariance of $\mathbf{y} = \mathbf{y}(t)$ into blocks corresponding to the unobserved (u) and observed (o) variables. The estimation of $\boldsymbol{\mu}_{\mathbf{y}}$ and $\Sigma_{\mathbf{y}}$ are important problems in practice, which were addressed in [9] for the case of network monitoring. For the purpose of this work, we shall assume that $\mu_{\mathbf{y}}$ and $\Sigma_{\mathbf{y}}$ are known. Assuming multivariate normality, we also have[2]

$$\boldsymbol{y}_u|\boldsymbol{y}_o \sim \mathcal{N}(\widehat{\boldsymbol{y}}_u, \Sigma_{uu} - \Sigma_{uo}\Sigma_{oo}^{-1}\Sigma_{ou}), \tag{2}$$

where $\widehat{\boldsymbol{y}}_u$ is given by (1). In this case, the BLUP $\widehat{\boldsymbol{y}}_u \equiv \mathbb{E}(\boldsymbol{y}_u|\boldsymbol{y}_o)$ is also the mean square optimal predictor.

Let the error covariance matrix be

$$\begin{aligned}\Sigma_{\rm err} \equiv \Sigma_{\rm err}(\mathcal{O}) &:= \mathbb{E}[(\widehat{\boldsymbol{y}}_u - \boldsymbol{y}_u)(\widehat{\boldsymbol{y}}_u - \boldsymbol{y}_u)^T|\boldsymbol{y}_o] \\ &= \mathbb{E}[(\widehat{\boldsymbol{y}}_u - \boldsymbol{y}_u)(\widehat{\boldsymbol{y}}_u - \boldsymbol{y}_u)^T] \\ &= \Sigma_{uu} - \Sigma_{uo}\Sigma_{oo}^{-1}\Sigma_{ou},\end{aligned} \tag{3}$$

where for the second equality we used the fact that for Gaussian random variables the error of estimation is independent to any linear or non-linear functional of the observations $\boldsymbol{y}_o$. The objective functions considered in this study are:

(i) *A-optimality:*

$$\mathrm{trace}(\Sigma_{\rm err}) \equiv \sum_{l\in\mathcal{U}} \mathrm{Var}(y_l|\boldsymbol{y}_o) = \mathbb{E}\|\widehat{\boldsymbol{y}}_u - \boldsymbol{y}_u\|^2, \tag{4}$$

where $\|\cdot\|$ stands for the Euclidean norm;

(ii) *E-optimality:*

$$\rho(\Sigma_{\rm err}) = \|\Sigma_{\rm err}\|_2, \tag{5}$$

where $\|\cdot\|_2$ stands for the spectral matrix norm, i.e., largest eigenvalue of $\Sigma_{\rm err}$. Then, the *optimal monitoring design* problem is given by:

Problem (Optimal Monitoring Design) *Find the optimal set $\mathcal{O}^* \subseteq \mathcal{L} := \{1,2,\ldots,L\}$ such that:*

$$Z(\mathcal{O}^*) = \min_{\mathcal{O}\subseteq\mathcal{L},\ s.t.\ |\mathcal{O}|=K} f(\Sigma_{\rm err}(\mathcal{O})), \tag{6}$$

*where $Z(\mathcal{O})$ is the prediction error when monitoring the set of links $\mathcal{O}\subseteq\mathcal{L}$ and $f(\cdot)$ is the optimality criterion (*e.g. trace *or* spectral norm*).*

[2]Henceforth, we shall assume that the matrix A has full row rank. Otherwise, our results can be shown to hold *mutatis mutandis* with $\Sigma_{oo}^{-1} = (A_o A_o^T)^{-1}$ viewed as the Moore-Penrose generalized inverse.

The *greedy heuristic* is a well-known fast method for solving (6). Starting from an empty set $\mathcal{O}$, this heuristic amounts to incrementally adding to $\mathcal{O}$ the link that minimizes the prediction error Z (or equivalently, maximizes the error reduction). Let $\delta_j(\mathcal{O}) = Z(\mathcal{O}) - Z(\mathcal{O}\bigcup\{j\})$ be the error reduction when adding element j to the set $\mathcal{O}$. The formal algorithm due to Nemhauser *et al.* [16] is as follows.

Greedy Heuristic.

1) *Let $\mathcal{O}^0 = \emptyset$, $\mathcal{N}^0 = \mathcal{L}$ and set $k = 1$.*
2) *At iteration k, select $i_k \in \mathcal{N}^{k-1}$ such that*

$$i_k \in \arg\max_{i\in\mathcal{N}^{k-1}} \delta_i(\mathcal{O}^{k-1}) \tag{7}$$

with ties settled arbitrarily.
3) *If $\delta_{i_k}(\mathcal{O}^{k-1}) \le 0$ then stop. Otherwise, set $\mathcal{O}^k = \mathcal{O}^{k-1}\bigcup\{i_k\}$ and $\mathcal{N}^k = \mathcal{N}^{k-1}\setminus\{i_k\}$.*
4) *If $k = K$ stop and output $\mathcal{O}^K$. Otherwise, set $k = k+1$ and go to step 2).*

The "naïve" implementation of the greedy heuristic is not feasible for large-scale networks, since our optimization criteria involve the inversion and multiplication of matrices with dimensions in the order of several tens of thousands. In our problem, however, there is a natural geometric structure related to PCA that can be used for developing fast heuristics. It also leads to an exact and efficient implementation of the classical greedy heuristic which avoids matrix inversions. The connections to PCA also yield lower bounds on the error in (6), which are of independent interest. We present these new results in Section IV, following the introduction of our working framework for network monitoring.

B. Large-scale Network Monitoring

Consider a communication network of N nodes and L links. The total number of traffic flows, i.e. source and destination (S,D) pairs, is denoted by J. Traffic is routed over the network along predefined routes described by a routing matrix $R = (r_{\ell,j})_{L\times J}$, with $r_{\ell,j} = 1$, when route j uses link ℓ and 0 otherwise. Let $\boldsymbol{x}(t) = (x_j(t))_{j=1}^J$ and $\mathbf{y}(t) = (y_\ell(t))_{\ell=1}^L$, $t = 1,2,\cdots$ be the vector time series[3] of traffic traversing all J routes and L links, respectively. We shall ignore network delays and adopt the assumption of *instantaneous propagation*. This is reasonable when traffic is monitored at a time-scale coarser than the round-trip time of the network, which is the case in our setting. We thus obtain that the link and route level traffic are related through the fundamental *routing equation*

$$\mathbf{y}(t) = R\boldsymbol{x}(t). \tag{8}$$

[3]Here, time is discrete and traffic loads are measured in bytes or packets per unit time, over a time scale greater than the round-trip time of the network.

It is well-known that single link/route traffic traces exhibit burstiness over time, which can be statistically explained via the notions of *long-range dependence* and *self-similarity* (see *e.g.* [26]).

In [8], we proposed a global mechanistic model for the traffic on an entire network. The traffic flow along each route was represented as a composition of multiple long-range dependent On/Off traces describing the behavior of individual "users". Such On/Off processes have been popular and successful models for the pattern of traffic generated by various protocols, services and applications (*e.g.*, peer-to-peer, file transfers, VoIP, http, etc.). It is well known that the composition of multiple independent traces of this type yields long-range dependent models, that are well-approximated by *fractional Guassian noise* (see *e.g.* [26]). On the other hand, using NetFlow data, we found that the traffic flows $x_j(t)$ across different routes j are relatively weakly correlated[4] (in j). Thus, the *routing* (8), becomes a primary cause of statistical dependence between the traffic traces $y_\ell(t)$ across different links ℓ in the network. In particular, the greater the number of common flows that pass through two given links, the greater the correlation between the traffic loads on these links. The dependence of $y_\ell(t)$ across "space" (links) ℓ and time t can be quantified and succinctly described in terms of the *functional fractional Brownian motion* (see [8]).

In [9], we developed further statistical methodology that allows one to estimate (using NetFlow data) the structure of the means $\boldsymbol{\mu_x} = \mathbb{E}\boldsymbol{x}(t)$, the flow-covariances $\Sigma_{\boldsymbol{x}} := \mathbb{E}(\boldsymbol{x}(t) - \boldsymbol{\mu_x})(\boldsymbol{x}(t) - \boldsymbol{\mu_x})^T$ and their relationship for all routes in the network. This leads to a practical factor model for the link loads $\mathbf{y}(t)$, which can be estimated *online* from traffic measurements of just a few links. The estimated model can in turn be used to perform *network prediction*, which is the topic of this paper.

IV. Principal Component Analysis

A. A geometric view of optimal prediction

We discuss next how our problem (6) relates to PCA. The covariance matrix $\Sigma_{\boldsymbol{y}}$ could be obtained via the routing equation (8) and the statistical characteristics of the J flows, i.e. $\Sigma_{\boldsymbol{y}} = R\Sigma_{\mathbf{x}}R^T$ or could be directly available through historical data from past link measurements (for more details on estimating the covariance see [9]). Without loss of generality, we decompose $\Sigma_{\mathbf{y}}$ using *singular value decomposition*, as $\Sigma_{\boldsymbol{y}} = AA^T$ with some $L \times J$ matrix A. The row-vectors of the matrix A will be denoted as $\boldsymbol{a}_\ell \in \mathbb{R}^J,\ \ell = 1, \cdots, L$.

For convenience, we let $\Sigma_{\rm err}(\mathcal{O}) = \mathbb{E}(\boldsymbol{y} - \widetilde{\boldsymbol{y}})(\mathbf{y} - \widetilde{\boldsymbol{y}})^T$, with $\boldsymbol{y} = (\boldsymbol{y}_u\ \boldsymbol{y}_o)^T$ and $\widetilde{\boldsymbol{y}} = (\widehat{\boldsymbol{y}}_u\ \boldsymbol{y}_o)^T$ be the error covariance matrix including both the observed and unobserved links. Using (3) one can show that $\Sigma_{\rm err}(\mathcal{O}) = \Sigma_{\boldsymbol{y}} - \Sigma_o\Sigma_{oo}^{-1}\Sigma_o^T$, where $\Sigma_o = AA_o^T,\ \Sigma_{oo} = A_oA_o^T$, and $A_o = (a_{ij})_{i\in\mathcal{O}, j\in\mathcal{J}}$ is a submatrix of A with rows corresponding to the set of observed links $\mathcal{O}$. In practice, $\Sigma_{\boldsymbol{y}}$ is typically non-singular.

One thus obtains:

$$\begin{aligned}\Sigma_{\rm err}(\mathcal{O}) &= A(I_J - A_o^T(A_oA_o^T)^{-1}A_o)A^T \\ &= A(I_J - P_{\mathcal{O}})A^T,\end{aligned} \tag{9}$$

where $P_{\mathcal{O}} := A_o^T(A_oA_o^T)^{-1}A_o$. The matrix $P_{\mathcal{O}}$ is the projection matrix onto the space $W_{\mathcal{O}} := \mathrm{Range}(A_o^T)$ spanned by the row vectors $\{\boldsymbol{a}_i : \boldsymbol{a}_i \in \mathbb{R}^J, i \in \mathcal{O}\}$ of A_o. Therefore, $(I_J - P_{\mathcal{O}})$ is the projection matrix onto the orthogonal complement $\mathrm{Range}(A_o^T)^\perp$.

Appendix A shows that $\|A - AP_{\mathcal{O}}\|_F^2 = \mathrm{trace}(A(I_J - P_{\mathcal{O}})A^T)$ and $\|A - AP_{\mathcal{O}}\|_2^2 = \rho(A(I_J - P_{\mathcal{O}})A^T)$, where $\|\cdot\|_F$ denotes the Frobenius[5] norm. These facts together with (9) imply that the problem in (6) is no different than the combinatorial problem:

$$\mathcal{O}^* := \underset{\mathcal{O}\subseteq\{1,\cdots,L\}, |\mathcal{O}|=K}{\arg\min} \|A - AP_{\mathcal{O}}\|_\xi^2, \tag{10}$$

where $\|\cdot\|_\xi$ is the spectral norm for $\xi = 2$ and the Frobenius norm for $\xi = F$.

Problem (10), however, has a nice *geometric* interpretation. It seeks the "optimal" subspace $W_{\mathcal{O}} := \mathrm{Range}(A_o^T)$ such that the distances, under the Frobenius[6] or spectral norm, of the matrix A to its row-wise projection $AP_{\mathcal{O}}$ are minimized.

B. PCA-based lower bounds

Observe that in (10), projection is restricted to the subspaces spanned by subsets of K rows of A. Should one relax this constraint and optimize over arbitrary K-dimensional subspaces of $\mathbb{R}^J$, one would achieve a lower bound for the objective function. In this case, PCA analysis shows that the optimal space W^* is given by $P_K = V_KV_K^T$ where $V_K = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K)$ is the matrix of the K principal eigenvectors of A^TA. Let $A^TA = V^TDV$ be the singular value decomposition (SVD) of A^TA with $D = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_K, ..., \lambda_J)$, where $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_K \geq \cdots \geq \lambda_J \geq 0$ (see [27], [3] and Appendix A). The above PCA-geometric observation readily gives the following *lower bounds* for the prediction error (see Appendix A for proofs).

Theorem 1 (PCA lower bound for trace)**.**

$$\sum_{i=K+1}^{J} \lambda_i = \|A - AP_K\|_F^2 \leq \mathrm{trace}(\Sigma_{\rm err}(\mathcal{O})). \tag{11}$$

[4]Except in periods of congestion where the TCP feedback mechanism induces dependence between the *forward* and *reverse* flows (see [8] and also [7]).

[5]By definition $\|B\|_F := \sqrt{\mathrm{trace}(BB^T)}$.

[6]The notion of distance is even more transparent in the Frobenius case – see Relation (32) in Appendix A.

Theorem 2 (PCA lower bound for spectral norm)**.**

$$\lambda_{K+1} = \|A - AP_K\|_2^2 \leq \rho(\Sigma_{\rm err}(\mathcal{O})). \qquad (12)$$

The geometric structure of the problem also suggests efficient heuristics, discussed in detail in the next section. For example, we can think of a sequential "greedy" method that picks the space $W_{\mathcal{O}}$ that has smallest *"angle"* with the space W^*. Note though that the spaces $W_{\mathcal{O}}$ in (10) should be spanned by K vectors $\boldsymbol{a}_i$, $i \in \mathcal{L}$ (*i.e.*, corresponding to K observed links). Thus, in principle, the PCA-based lower bounds are strict, yet extremely useful since these bounds also hold for the exact optimal solution $\mathcal{O}^*$. Therefore, a small relative gap between the error of a heuristic and the PCA lower bound implies a good approximation to the value $Z(\mathcal{O}^*)$ of the optimal solution.

V. Approximation Algorithms

A naïve implementation of the described greedy algorithm for both[7] A- and E-optimality criteria has a complexity of $O(KL^4)$. This is because operations such as matrix inversion, matrix multiplication and the calculation of the trace or spectral norm are involved whenever $\delta_i(\mathcal{O}^{k-1})$ is calculated (see (7) and Eqs.(2)-(5)). This section presents fast, approximation algorithms that substantially reduce this computational complexity (see Fig. 7 to grasp an idea of the speedup achieved).

Motivated by the discussion at the end of the previous section, one idea is to first pick a link $i_1 \in \mathcal{L}$ for which the vector $\boldsymbol{a}_{i_1}$ is "closest" to the first PCA component $\mathbf{v}_1$. If $K = 1$, the procedure ends, and one can show – by (9) and Theorem 3 of the sequel – that this selection is very close to the *optimal* choice where the notions of "close" and "optimal" depend on the type of norm or optimality criterion. If $K > 1$, we "subtract" the effect of the chosen link i_1 by considering only the orthogonal projections of the $\boldsymbol{a}_i$'s for the remaining links onto the space $\text{span}\{\boldsymbol{a}_{i_1}\}^{\perp}$. We then construct a new matrix A with rows given by the above projections and repeat the procedure iteratively K times.

Formally, the matrix A is updated as follows. Assume the iterative procedure selects link i_k at step k to be added to the set of selected links $\mathcal{O}$. We update A with the rule

$$A^{(k)} = A^{(k-1)}\Big(I_J - \frac{\boldsymbol{a}_{i_k}^{(k-1)}\boldsymbol{a}_{i_k}^{(k-1)^T}}{\|\boldsymbol{a}_{i_k}^{(k-1)}\|^2}\Big), \qquad (13)$$

with $A^{(0)} = A$ and the row-vectors of $A^{(k-1)}$ being $\boldsymbol{a}_i^{(k-1)} \in \mathbb{R}^J, i \in \{1, 2, \ldots, L\}$. The following proposition justifies this method, by establishing that the proposed procedure can be used to sequentially update the prediction error covariance $\Sigma_{\rm err}(\mathcal{O})$. This sequential projection property is the key behind the computational efficiency of the proposed heuristics presented in the sequel, since it allows us to avoid expensive operations such as matrix inversions. Its proof is given in Appendix B.

Proposition 1 (Sequential Computation of $\Sigma_{\rm err}(\mathcal{O})$)**.** Let $\mathcal{O}^k$ be the set of selected links at the end of step k, and $A^{(k)}$ the iteratively updated matrix, as shown in Eq. (13). Then,

$$A^{(k)}A^{(k)^T} = \Sigma_{\rm err}(\mathcal{O}^k) \text{ where} \qquad (14)$$
$$\Sigma_{\rm err}(\mathcal{O}^k) \equiv A(I_J - A_{o_k}^T(A_{o_k}A_{o_k}^T)^{-1}A_{o_k})A^T.$$

A. A-optimality (trace) criterion

In our first heuristic[8], we implement the geometric idea discussed above where, at step k, we select the link $i_k \in \mathcal{L}$ whose row-vector $\boldsymbol{a}_{i_k}$ is closest to the principal component of the current version of A. We can choose between two options for vector proximity; the smallest angle or the longest projection. We decided to work with the latter. We obtain the principal component using the *power method* [27]. The steps of the algorithm are:

Algorithm 1 (PCA Projection Heuristic - PCAPH)**.** *Let $\mathcal{O}^0 = \emptyset$ and $A^{(0)} = A$. Set $k = 1$.*

1) Power Method Step: *Using the* power method *obtain a fast approximation to the principal eigenvector $\mathbf{v}_1$ of $A^{(k-1)^T}A^{(k-1)}$.*
2) Selection: *At iteration k, choose:*

$$i_k \in \underset{i \in \mathcal{L}\setminus\mathcal{O}^{k-1}}{\arg\max} |\mathbf{v}_1^T\boldsymbol{a}_i^{(k-1)}|$$

where $\boldsymbol{a}_i^{(k-1)} \in \mathbb{R}^J, i \in \{1, 2, \ldots, L\}$ are the row-vectors of $A^{(k-1)}$. Put i_k in the list of links to be monitored, i.e., $\mathcal{O}^k := \mathcal{O}^{k-1}\bigcup\{i_k\}$.
3) Projection/Error Reduction: *The rows of the matrix $A^{(k)}$ are the orthogonal projections of the rows of $A^{(k-1)}$ onto $(\text{span}\{\boldsymbol{a}_{i_k}^{(k-1)}\})^{\perp}$. Formally,*

$$A^{(k)} := A^{(k-1)}\Big(I_J - \frac{\boldsymbol{a}_{i_k}^{(k-1)}\boldsymbol{a}_{i_k}^{(k-1)^T}}{\|\boldsymbol{a}_{i_k}^{(k-1)}\|^2}\Big). \qquad (15)$$

4) *Set $k = k + 1$. If $k < K$, go to step 1).*

As shown in Section VII, this strategy usually yields a monitoring design with slightly larger prediction error than the greedy strategy, but is orders of magnitude faster in execution time. Specifically, step 1) requires $O(mJL)$ computations, where m is the number of iterations the power method executes ($m \ll \min\{L, J\}$). We need $O(LJ)$ computations per iteration for the matrix-vector multiplication in the power method loop. Steps 2) and 3) require $O(LJ)$ operations.

[7] A more meticulous calculation shows that the complexity for A-optimality is $O(K^2L^3)$.

[8] This heuristic is for both A- and E-optimality. For paper organization purposes, it is placed under A-optimality.

Lemma 1. *The complexity of the PCA Projection Heuristic is* $O(mKLJ)$.

The next algorithm, is a very fast implementation of the classical greedy algorithm. It avoids calculating the inverse of the covariance matrix Σ_{oo}. Instead, at each iteration we seek the column vector that maximizes the error reduction. This is equivalent to finding the vector that maximizes the squares of the projections of the remaining vectors onto itself. For A-optimality, one can show that the *error reduction* at each step k, given that links $\{i_1, ..., i_{k-1}\}$ were already chosen, is equal to:

$$R_k(i) := R_{\{i_1,\ldots,i_{k-1}\}}(i) = \sum_{j=1}^{L} \frac{|\boldsymbol{a}_j^{(k-1)^T} \boldsymbol{a}_i^{(k-1)}|^2}{\|\boldsymbol{a}_i^{(k-1)}\|^2}, \tag{16}$$

with $i \in \mathcal{L} \setminus \{i_1, ..., i_{k-1}\}$. In words, it equals the sum of the squares of the projections to space $\text{span}\{\boldsymbol{a}_i^{(k-1)}\}$. This step is accomplished by the first step of the algorithm. The second step, updates the matrix $A^{(k)}$.

Algorithm 2 (Fast Greedy Exact - FGE)**.** *Let* $\mathcal{O}^0 = \emptyset$ *and* $A^{(0)} = A$. *Set* $k = 1$.

1) SELECTION: *At iteration* k, *choose:*

$$i_k \in \underset{i \in \mathcal{L} \setminus \mathcal{O}^{k-1}}{\arg\max} \sum_{j=1}^{L} \frac{|\boldsymbol{a}_j^{(k-1)^T} \boldsymbol{a}_i^{(k-1)}|^2}{\|\boldsymbol{a}_i^{(k-1)}\|^2} \tag{17}$$

where $\boldsymbol{a}_i^{(k-1)} \in \mathbb{R}^J, i \in \{1, 2, \ldots, L\}$ *are the row vectors of* $A^{(k-1)}$. *Set* $\mathcal{O}^k = \mathcal{O}^{k-1} \bigcup \{i_k\}$.

2) PROJECTION/ERROR REDUCTION: *Do step 3) of Algorithm 1, i.e.,*

$$A^{(k)} = A^{(k-1)} \Big(I_J - \frac{\boldsymbol{a}_{i_k}^{(k-1)} \boldsymbol{a}_{i_k}^{(k-1)^T}}{\|\boldsymbol{a}_{i_k}^{(k-1)}\|^2} \Big). \tag{18}$$

3) Set $k = k + 1$. *If* $k < K$, *go to step 1).*

Step 1) is a "greedy step" since it picks the link that reduces the error the most. It requires $O(JL^2)$ operations while, step 2) requires $O(LJ)$ operations after suitably rearranging the order of operations.

Lemma 2. *The computational complexity of the Fast Greedy Exact algorithm is* $O(KJL^2)$.

Remark 1. *Algorithm PCAPH relies on the fast performance of the power method, whose convergence speed depends on the ratio* $|\lambda_2|/|\lambda_1|$ *of the matrix under study. Thus, one might think that the performance of PCAPH may deteriorate when that ratio is close to* 1. *However, it is not affected because: a) We are only interested to an approximation of the principal eigenvector so a few iterations of the power method suffice, and b) other iterative methods could be used instead, such as the Rayleigh quotient iteration [27] that has a cubic convergence speed when an approximate eigenvector is provided (say, from the power method).*

B. E-optimality (spectral norm) criterion

For E-optimality, we present a very fast implementation of the greedy heuristic. Relation (9) and Proposition 1 allow us to avoid computationally expensive operations like matrix inversion and singular value decomposition, which leads to drastic improvements in performance. The next algorithm was motivated by the following characterization of the largest eigenvalue [27]:

$$\rho(\Sigma_{\text{err}}) := \lambda_1(\Sigma_{\text{err}}) = \max_{\boldsymbol{z} \in \mathbb{R}^L, \|z\|=1} \boldsymbol{z}^T \Sigma_{\text{err}} \boldsymbol{z}. \tag{19}$$

Algorithm 3 (Fast Greedy Randomized - FGR)**.** *Let* $\mathcal{O}^0 = \emptyset$ *and* $A^{(0)} = A$. *Set* $k = 1$.

1) INITIALIZATION: *Generate* m *independent, normally distributed random vectors* $\mathbf{x}_i \in \mathbb{R}^L, i = 1, 2, \ldots, m$ *from* $\mathcal{N}(0, I_L)$ *and set* $\mathbf{z}_i := \mathbf{x}_i / \|\mathbf{x}_i\|$.

2) SAVINGS STEP: *At iteration* k, $k = 1, 2, \ldots, K$, *calculate:*

$$c_i := (\mathbf{z}_i^T A^{(k-1)}) \cdot (A^{(k-1)^T} \mathbf{z}_i), \forall i = 1, 2, \ldots, m. \tag{20}$$

3) SELECTION: *At iteration* k, *select:*

$$j_k \in \underset{j \in \mathcal{L} \setminus \mathcal{O}^{k-1}}{\arg\min} \Big\{ \max_{\mathbf{z}_i} [c_i - \frac{\mathbf{z}_i^T \mathbf{b}_j^{(k-1)} \mathbf{b}_j^{(k-1)^T} \mathbf{z}_i}{\|\boldsymbol{a}_j^{(k-1)}\|^2}] \Big\}, \tag{21}$$

where $\mathbf{b}_j^{(k-1)} := A^{(k-1)} \boldsymbol{a}_j^{(k-1)}$ *and* $\boldsymbol{a}_j^{(k-1)} \in \mathbb{R}^J, j \in \{1, 2, \ldots, L\}$ *are the column vectors of* $A^{(k-1)^T}$. *This corresponds to finding the link* j *that minimizes the error* $\|A^{(k)}(I - \boldsymbol{a}_j \boldsymbol{a}_j^T / \|\boldsymbol{a}_j\|^2) A^{(k)^T}\|_2$. *Set* $\mathcal{O}^k = \mathcal{O}^{k-1} \bigcup \{i_k\}$.

4) PROJECTION/ERROR REDUCTION: *Do step 3) of Algorithm 1.*

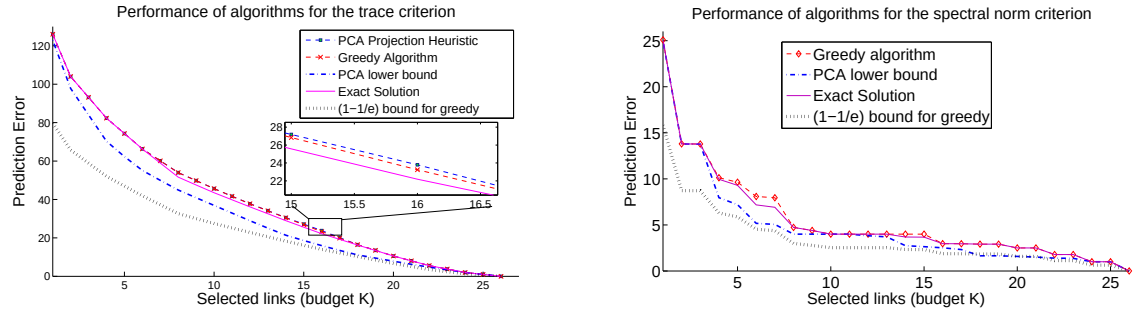
5) Set $k = k + 1$. *If* $k < K$, *go to step 2).*

In step 1), we randomly sample m unit vectors $\mathbf{z}_i$: $\mathbf{z}_i \in \mathbb{R}^L, i = 1, 2, \ldots, m, \|\mathbf{z}_i\| = 1$. We use these vectors in step 3) to approximate the maximum in (19) and hence the largest eigenvalue. Note, that in step 2) we save the c_i values so as to omit unnecessary repetitions of the same quantity. In step 3), we also choose the vector (link) that minimizes the error expressed through the largest eigenvalue. Finally, in step 4), we update matrix A for use in the next iteration. Step 2) requires $O(mLJ)$ operations and step 3) $O(mLJL)$.

Lemma 3. *The computational complexity of the Fast Greedy Randomized algorithm is* $O(mKL^2J)$.

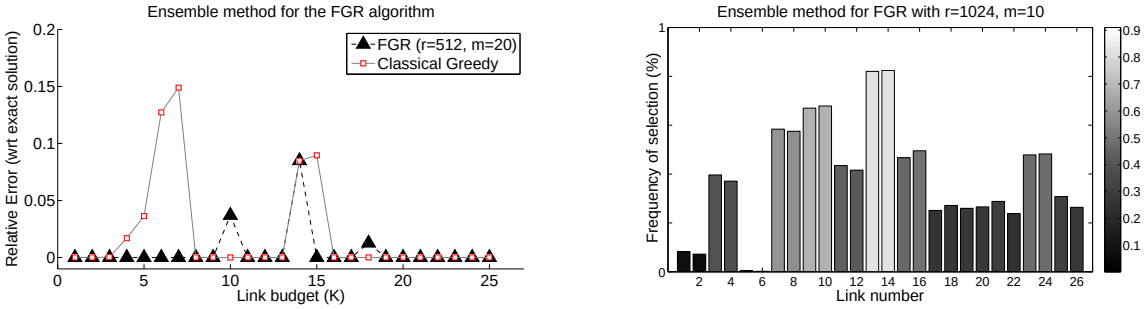
The following propositions present theoretical bounds on the quality of approximating $\tilde{\lambda}_1$, the largest eigenvalue of a matrix, say Σ. For proofs see the Appendix.

Proposition 2. Let λ_1 be the true eigenvalue and $\tilde{\lambda}_1$ the approximated one using the heuristic described in Algorithm 3. Let also m be the number of random



(a) A-optimal criterion (b) E-optimal criterion

Fig. 2. Evaluation of approximation algorithms on Internet2, and comparison with the exact optimal solution. Moreover, illustration of the PCA lower bound, which is useful for assessing the quality of approximation when the exact solution is not known. Compare it with the loose (1-1/e) bound proposed by Nemhauser/Wolsey [15]. Of course, the bound of [15] cannot be claimed, due to absence of submodularity.



(a) Relative error (wrt optimal solution) (b) Link selection (for $K = 10$)

Fig. 3. Ensemble (distributed implementation) of FGR algorithm for Internet2 network. (Left) Ensemble of FGR yields the exact optimal solution 88% of the time, whereas classical greedy 76%. (Right) Link selection frequency. Links that connect West to East sites (*e.g.* links 13, 14 and 9, 10 that correspond to KANS–CHIC and SALT–KANS, see Table I) are selected the most often by our randomized algorithm. Information from links 5, 6 is redundant, so these links are almost never selected.

vectors $\mathbf{z}_i \in \mathbb{R}^L$ used in the heuristic. Then, for any positive scalar $\varepsilon > 0$, the following holds:

$$\mathbf{P}\Big(\frac{\tilde{\lambda}_1}{\lambda_1} > 1-\varepsilon\Big) \geq 1 - e^{-\delta(m,\varepsilon,L)}, \qquad (22)$$

where

$$\delta(m,\varepsilon,L) = \frac{2m\Gamma(L/2)}{(L-1)\sqrt{\pi}\Gamma(-1/2+L/2)}\sqrt{\frac{\varepsilon^{L-1}}{1-\varepsilon}}, \qquad (23)$$

with $\Gamma(\cdot)$ being the Γ function.

When some extra information for the eigenvalues of matrix Σ is available, the next proposition can be more elucidating regarding the value of m. Such information might be obtained by computing the eigenvalues *once*, before the algorithm starts, so as to help the designer grasp an idea for adequate values for m.

Proposition 3. Let λ_1 be the true eigenvalue, and $\tilde{\lambda}_1$ the approximated one using the heuristic described in Algorithm 3. Let also m be the number of random vectors $\mathbf{z}_i \in \mathbb{R}^L$ used in the heuristic. The eigenvalues of matrix Σ are $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_L$, and $c_1 \geq c_2 \geq \ldots c_L$ with $c_i = \lambda_i/\lambda_1$ the normalized values thereof. For any positive scalar $\varepsilon > 0$, there exists an index t such that $c_i > 1-\varepsilon$, $\forall i \leq t$ and $c_i \leq 1-\varepsilon$ otherwise. Then,

$$\mathbf{P}\Big(\frac{\tilde{\lambda}_1}{\lambda_1} > 1-\varepsilon\Big) \geq 1 - [F_{t,L-t}\Big(\frac{L-t}{t}\frac{1-\varepsilon}{c_t-(1-\varepsilon)}\Big)]^m, \qquad (24)$$

where $F_{\nu_1,\nu_2}(\cdot)$ is the cumulative distribution function for the F-distribution with parameters ν_1 and ν_2.

C. Ensemble methods for randomized algorithms

Algorithm FGR can be characterized as a *randomized algorithm*. It uses m random vectors to approximate the largest eigenvalue λ_1, needed to accomplish the *Selection* step. PCAPH may also be implemented in a randomized fashion; since it involves the power method for approximating $\mathbf{v}_1$, one can utilize a random vector for the method's initial value. This randomization gives rise to the idea of ensembling.

The main idea of the *ensemble* method is to pick a small m – which makes the algorithm much faster – and run several, say r, independent instances of the algorithm, in parallel. We then select the solution set that yields the minimum prediction error among the ensemble. Note that unlike the greedy approach, the resulting solution sets $\mathcal{O}^K$ are no longer nested, i.e. some links may be excluded from $\mathcal{O}^K$ and replaced with others as K grows. This helps avoiding one of the artificial constraints that greedy procedures impose on the solution sets.

TABLE I
ID'S OF THE 26 LINKS OF THE INTERNET2 NETWORK. ODD LINK ID'S CORRESPOND TO THE UPLINK DIRECTION AND THE EVEN TO DOWNLINK; I.E. LINK 7 IS THE LOS ANGELES TO HOUSTON LINK AND LINK 8 IS THE HOUSTON TO LOS ANGELES LINK.

Link ID	Origin → Destination
1,2	Los Angeles → Seattle
3,4	Seattle → Salt Lake City
5,6	Los Angeles → Salt Lake City
7,8	Los Angeles → Houston
9, 10	Salt Lake City → Kansas City
11, 12	Kansas City → Houston
13, 14	Kansas City → Chicago
15, 16	Houston → Atlanta
17, 18	Chicago → Atlanta
19, 20	Chicago → New York
21, 22	Chicago → Washington
23, 23	Atlanta → Washington
25, 26	Washington → New York

Our experiments with the topology of Fig. 8, indicate that for large enough r, ensemble methods can often come close and, in fact, yield the optimal solution of Problem (6). For example, Fig. 3(a) shows the results of a distributed implementation of FGR with $r = 512$ and $m = 20$. Comparing with a single execution of the classical greedy (or even FGR with $r = 1$, $m \gg 20$), we note that the exact optimal solution is obtained 88% of the time. The solution can be obtained in minutes, rather than hours, as needed when an integer programming formulation is used for the exact solution. Fig. 3(b) shows the selection frequency of each link when FGR is employed with $r = 1024, m = 10$. It is pleasing to observe that the links that bridge the East with the West sites of the topology have the highest probability of being included in the optimal solution.

Furthermore, as verified in Table II, randomization allows the network designer to adhere to smaller values for m when the option of parallel or distributed implementation of the algorithm is available. Table II shows the error of the FGR algorithm with respect to the optimal solution, for the Internet2 network and link budget $K = 14$. We ran several versions of FGR with different values for random vectors m and amount of parallelization r. The results were run in Matlab on an 8-core computer. Note that the advantages of parallelization are not entirely revealed in our example since, in most cases, r is much greater than the number of cores. Thus, it was rather a distributed implementation than a parallel one. Nonetheless, the proposed ensemble method is inherently parallelizable. Should enough resources are present, a very brief time is needed for computing a monitoring set with less prediction error than the "naïve" implementation of the greedy heuristic. For example, for $m = 64$ and $r = 256$ we get a relative error of 4.8%, whereas greedy gives 8.5%. If we had 256 cores available, then the computational time for that (m, r) pair would not exceed the 7 seconds that our 8-core machine needed for running FGR with $m = 512$, $r = 8$.

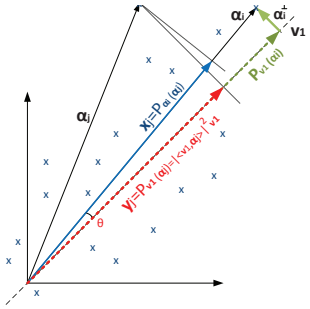


Fig. 4. Visualization of the geometric interpretation of PCA analysis. $P_{\boldsymbol{y}}(\boldsymbol{x})$ refers to the projection of vector $\boldsymbol{x}$ to vector $\boldsymbol{y}$.

VI. PERFORMANCE GUARANTEES FOR ERROR REDUCTION

This section introduces performance guarantees for the error reduction (see Eq. (16)) that can be achieved by algorithms PCAPH and FGE when the criterion is A-optimality. Note that the $(1 - 1/e)$ bounds for greedy algorithms developed by Nemhauser/Wolsey [15] cannot be claimed, since the submodularity property does not hold for our objective functions. For proofs see Appendix C.

Theorem 3. *Suppose that at iteration k the observed set $\mathcal{O}^k = \{i_1, ..., i_k\}$, and the resulting matrix is $A^{(k)}$. Let also $\boldsymbol{a}_i^{(k)}$ be a candidate vector for selection by the algorithm on step $k+1$. Then, the error reduction (see Eq.* (16)*) at step $k+1$ is bounded below as follows:*

$$\sum_{j=1}^{L} \frac{|\boldsymbol{a}_j^{(k)^T} \boldsymbol{a}_i^{(k)}|^2}{\|\boldsymbol{a}_i^{(k)}\|^2} \geq \tag{25}$$

$$\geq (\gamma^{(k)})^2 \lambda_1^{(k)} - 2\gamma^{(k)} \sqrt{1 - (\gamma^{(k)})^2} \sqrt{\lambda_1^{(k)}} \sqrt{\sum_{j=1}^{L} \lambda_j^{(k)}},$$

*where $\lambda_j^{(k)}$ is the jth largest eigenvalue of $A^{(k)^T} A^{(k)}$ (*i.e.*, after we have updated our matrix at the previous step k, see* (13)*), and $\gamma^{(k)}$ is the cosine (*i.e.*, see angle θ of Fig. 4) between the principal eigenvector $\mathbf{v}_1^{(k)}$ of $A^{(k)^T} A^{(k)}$ and the selected vector $\boldsymbol{a}_i^{(k)}$, i.e.*

$$\gamma^{(k)} = \frac{|\mathbf{v}_1^{(k)^T} \boldsymbol{a}_i^{(k)}|}{\|\boldsymbol{a}_i^{(k)}\|}. \tag{26}$$

Theorem 4. *Suppose that the conditions of Theorem 3 hold. Then, at iteration $k+1$, the error reduction is bounded from above as follows:*

TABLE II

ENSEMBLING FOR VARIOUS VALUES OF m AND r ON INTERNET2 NETWORK FOR $K = 14$. WE RAN 50 REPLICATIONS FOR EACH (m, r) PAIR TO OBTAIN THE RELATIVE MEAN ERROR (RME) W.R.T THE EXACT SOLUTION, AND 95% CONFIDENCE INTERVALS. CLASSICAL GREEDY'S RME IS 8.5%. THE EXPERIMENTS WERE RUN IN MATLAB ON AN 8-CORE COMPUTER.

(m,r)	32000,1	1024,4	512,8	512,16	256,16	128,32	64,256	32,512	32,256	8,512	4,1024	16,1024
95% CI (lb)	9.8	8.3	8.5	8.0	7.8	7.6	4.4	5.1	6.4	7.8	7.4	7.8
RME (%)	**10.2**	**8.4**	**8.5**	**8.2**	**8.0**	**7.8**	**4.8**	**5.4**	**6.6**	**8.0**	**7.6**	**8.0**
95% CI (ub)	10.6	8.6	8.5	8.3	8.1	8.0	5.1	5.7	6.9	8.1	7.8	8.1
Time (secs)	169	7.5	7.0	13.4	11.8	21.7	185.2	419.8	183.2	417.7	1021	1029

$$\sum_{j=1}^{L} \frac{|\boldsymbol{a}_j^{(k)^T} \boldsymbol{a}_i^{(k)}|^2}{\|\boldsymbol{a}_i^{(k)}\|^2} \leq \lambda_2^{(k)} + (\gamma^{(k)})^2 \lambda_1^{(k)} + \quad (27)$$

$$+2\gamma^{(k)} \sqrt{1-(\gamma^{(k)})^2} \sqrt{\lambda_1^{(k)}} \sqrt{\sum_{j=1}^{L} \lambda_j^{(k)}},$$

*where $\lambda_j^{(k)}$ is the jth largest eigenvalue of $A^{(k)^T} A^{(k)}$ (*i.e.*, after we have updated our matrix at the previous step k, see* (13)*), and $\gamma^{(k)}$ is as defined in Eq.* (26)*.*

Fig. 5 demonstrates the bounds when the FGE algorithm is run on a network with $N = 100$ nodes and $L = 195$ links (see section VII-C for more details on that network).

Remark 2 (Interpretation of Theorems 3 and 4)**.** *Say that the algorithm selects the link that corresponds to vector $\boldsymbol{a}_i^{(k)}$ which is at the same direction as $\mathbf{v}_1^{(k)}$. Then, $\gamma^{(k)} = 1$ and our bounds tell us that we will have a significant reduction, with value between $\lambda_1^{(k)}$ and $\lambda_1^{(k)} + \lambda_2^{(k)}$. On the other hand, if the algorithm selects a vector with $\gamma^{(k)} = 0$, then the best reduction we should expect to achieve is $\lambda_2^{(k)}$. This is because we are essentially selecting from vectors that are perpendicular to $\mathbf{v}_1^{(k)}$, and the "favorable choice" would be some vector that is co-directional with the second principal axis $\mathbf{v}_2^{(k)}$. For all other values of $\gamma^{(k)}$, the error reduction is in between the values given by the theorems, taking into account, of course, that it should be non-negative and not greater than the error reduction given by the PCA bound of Theorem 1.*

Remark 3 (E-optimality bounds)**.** *For E-optimality error reduction bounds we can claim the result of Theorem 2. This says, that at iteration $k+1$ we can expect an error reduction at most equal to $\lambda_1^{(k)}$ (see proof of the aforementioned theorem in Appendix A). In addition, we have the trivial lower bound that error reduction should be non-negative. At this moment, existence of tighter bounds remains an open problem.*

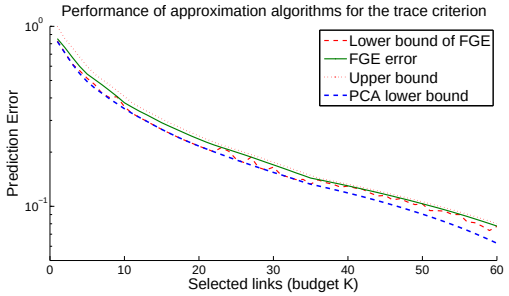


Fig. 5. Illustration of lower and upper bounds offered by Theorems 3 and 4 for the $N = 100$ nodes, $L = 195$ links network.

VII. PERFORMANCE EVALUATION

This section evaluates our algorithms under a variety of scenarios[9]. We start with a short discussion for the choice of budget K, and then we perform various comparisons of the proposed algorithms for different network sizes and topologies. Then, we employ our approximation methods to a real-world kriging application, demonstrating that the efficient performance of our algorithms allows for online implementation even when network conditions, like link importance, rapidly change.

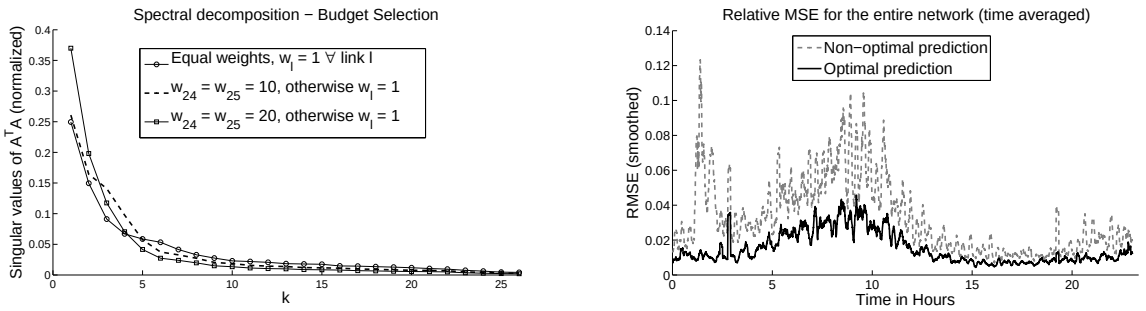
A. Choosing the "budget" K

Thus far we have assumed that the number of links to be monitored, namely the "budget" K, is known to the network operator. However, in practice, this is an unknown parameter dependent on the monetary budget available. We now address the question of choosing an adequate value for K, so that the network operator can spend the least amount of money while monitoring the network with sufficient accuracy. The answer comes from PCA via the spectral decomposition of the matrix $A^T A$. One wants to choose K according to the "spectrum" of the data. Specifically, K should be such that (see Section 6.1 [28] for more details and other criteria):

$$\sum_{i=1}^{K} \lambda_i \geq 0.80 \times \text{trace}(A^T A), \quad (28)$$

where $(\lambda_1 \geq \cdots \geq \lambda_p)$, $p = \min\{L, J\}$, are the eigenvalues of the matrix $A^T A$.

[9]Unless mentioned otherwise, all experiments were run in Matlab on a 2.4GHz, dual-core computer with 4GB memory.



(a) Budget selection

(b) Weighted RMSE.

Fig. 6. Kriging on Internet2. The weighted-error case. (Data for 03/17/09.) Note the steep error drop ($\approx 100\%$) between the two predictions around the 10th hour.

Fig. 6(a) shows the "spectra" of three signals, based on the Internet2 traffic data obtain on March 17, 2009. We plot the case of uniform priority, the case where links 24 and 25 are assigned weights $w_l = 10$, and the case where $w_l = 20$ for $l = 24, 25$. (More details on weighted link monitoring follow on subsection VII-F.) Note that as the importance of links increases the "energy" of the signal is concentrated on fewer singular values. This is intuitively appealing since it suggests that the prediction error will be reduced the most if we can afford a budget K that is at least as large as the number of the high importance links.

B. Greedy Vs. Exact algorithms

We use a real-world network, namely Internet2, to evaluate our approximation algorithms against the exact solution obtained via an exhaustive search. We also calculate and demonstrate the PCA lower bound. Internet2 (formerly Abilene) involves $L = 26$ links, $N = 9$ nodes and $J = 72$ routes (see [9], [11]). For simplicity, we assume that the flow-covariance matrix is $\Sigma_x = I_J$ (see also [8], [9]). In Fig. 2(a) we examine the trace criterion and in Fig. 2(b) the spectral norm one. In both cases, the exhaustive search required several hours to converge to a solution (the implementation was done in Matlab using CPLEX). On the contrary, the heuristics (PCAPH and greedy algorithms) terminate in a few seconds and yield a solution very close to the optimal one.

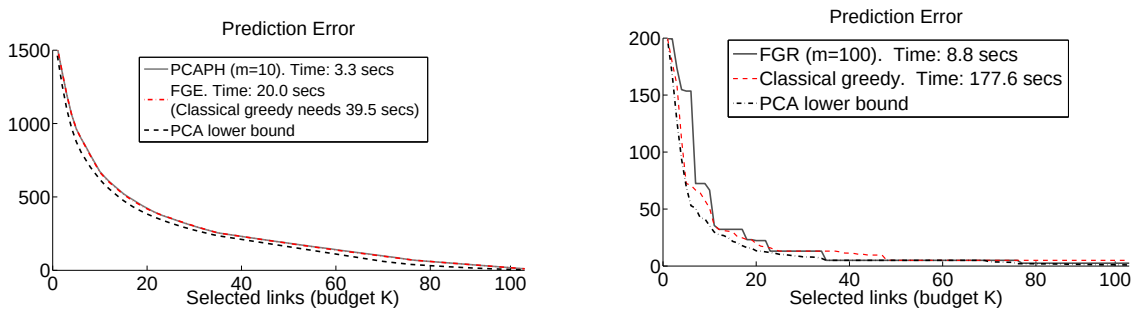
C. Comparisons of Approximation Algorithms

We next juxtapose the computational performance of our fast approximation algorithms against the classical greedy heuristic. We run our simulations on a moderate-sized network of $N = 100$ nodes generated using preferential attachment, as described in [29]. We assume J source-destination (S, D) flows with identical traffic, such that $x_j \sim \mathcal{N}(\mu, 1), j \in \{1, 2, ..., J\}$. The route for each (S, D) pair is chosen using the *Floyd–Warshall* shortest path routing algorithm. We created a network with a routing matrix R of $L = 195$ links and $J = 500$ flows.

Fig. 7(a) illustrates the results for A-optimality. Our proposed heuristics are notably faster than the naïve implementation of the greedy algorithm. Specifically, PCAPH is $10\times$ faster than classical greedy, and FGE is $2\times$ faster. The PCAPH algorithm significantly outperforms all other algorithms at the expense of having a slightly larger prediction error. Fig. 7(b) depicts the case of E-optimality. Again, our algorithm performs significantly faster ($20\times$ faster) than the naïve implementation of the greedy heuristic. We used $m = 100$ random vectors for the FGR algorithm. In both figures, we use the PCA lower bound to qualitatively assess the solutions of the algorithms, since obtaining the exact solution is not computationally feasible.

D. Scaling Up

In this section, we evaluate our algorithms in a large-scale network constructed via the topology simulation tool *Inet* [30]. The generated network has $N = 3050$ nodes, and we created a matrix A with $L = 4800$ links and $J = 1000$ flows. The naïve implementation of greedy was not executed for this scenario; based on its inferior computational performance observed earlier (see Fig. 7), such execution would be impractical. Its theoretical complexity is two orders of magnitude slower than our new heuristic and one order of magnitude slower than the efficient implementation of greedy. Table III shows the running time and the relative error of our algorithms with respect to the PCA lower bound. Observe that the PCAPH (ran with $m = 100$) and FGR ($m = 10$) terminate extremely fast to a near-optimal solution. Actually, FGR yields the optimal solution, which in this case attains the PCA lower bound. Thus, both algorithms are appealing and robust methods for *online* optimal monitoring design in large realistic networks.



(a) A-optimality (b) E-optimality

Fig. 7. Prediction error on a network with 100 nodes. Note the dramatic decrease in computation time between the naïve greedy and the proposed algorithms of this paper.

TABLE III
INET TOPOLOGY. $N = 3050$ NODES, $L = 4800$ LINKS. $K = 100$.

Heuristic	PCAPH	FGE	FGR
Time (secs)	54	2760 (46min)	546 (9.1min)
Rel. Err. (%)	1.03	0.99	0

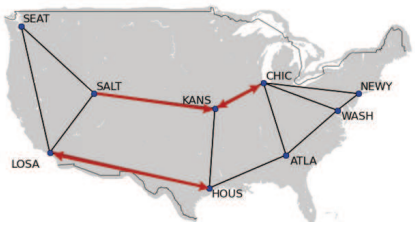


Fig. 8. Optimal monitoring on Internet2 network. The links in bold red lines are the "best" links to monitor. More details on this real-world application in Section VII.

E. Network Kriging in practice

We illustrate next the importance of optimal selection applied in the context of network prediction. We used the real-world data collected from the Internet2 network (see [9], [11]) on March 17, 2009. We utilized the PCAPH algorithm to calculate the optimal set of links to be monitored. Fig. 9(a) depicts the true traffic on link from CHIC (Chicago) to WASH (Seattle) and the predicted traffic when the optimal set of links is used. As Fig. 8 shows, this includes links KANS→CHIC, CHIC→KANS, LOSA→HOUS, HOUS→LOSA and SALT→KANS. As one would expect, an optimal global view is attained by monitoring links that connect network sites between West and East (see also [7]). Fig. 9(b) shows the empirical relative mean squared error (ReMSE) for the whole network on the day of interest. We compare the quality of prediction when monitoring: a) a non-optimally chosen set, b) the optimal set used throughout the day, and c) the optimal set, periodically recalculated every 8 hours. The last method accounts the newest history available, dynamically re-estimates matrix $\Sigma_{\mathbf{y}}$ and calculates the new optimal set. ReMSE is defined as,

$$ReMSE(t) = (\sum_{l\in\mathcal{U}}(\widehat{y}_l(t) - y_l(t))^2)/\sum_{l\in\mathcal{U}} y_l(t)^2).$$

The ReMSE time average is 0.09, 0.07 and 0.056, respectively. This clearly shows the advantages of using fast approximation algorithms that can be dynamically used for optimal link monitoring. In the following subsection, we show that the same algorithms can be employed for solving the *weighted* monitoring design problem, arising when the importance of monitoring sites is not uniform.

F. Weighted Link Monitoring

Our methods can be easily adjusted to handle cases where the relative importance of the links is not uniform. Such weighted design is particularly useful when network conditions drastically change in a dynamic manner. Recall that in the trace criterion case we have:

$$\begin{aligned}&\text{trace}(\mathbb{E}[(\widehat{\boldsymbol{y}}-\boldsymbol{y})(\widehat{\boldsymbol{y}}-\boldsymbol{y})^T])\\ &\quad=\mathbb{E}(\text{trace}[(\widehat{\boldsymbol{y}}-\boldsymbol{y})(\widehat{\boldsymbol{y}}-\boldsymbol{y})^T])\\ &\quad=\mathbb{E}[\|\widehat{\boldsymbol{y}}-\boldsymbol{y}\|^2]=\mathbb{E}[(\widehat{\boldsymbol{y}}-\boldsymbol{y})^T(\widehat{\boldsymbol{y}}-\boldsymbol{y})].\end{aligned} \quad (29)$$

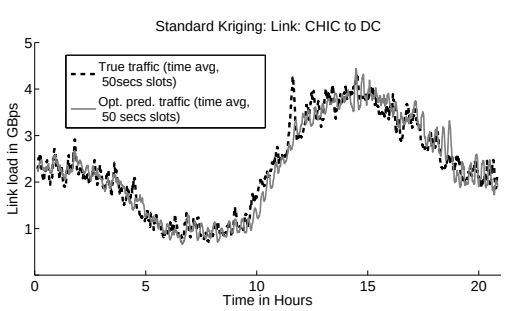
The weighted monitoring design problem aims to minimize

$$\mathbb{E}[(\widehat{\boldsymbol{y}}-\boldsymbol{y})^T G(\widehat{\boldsymbol{y}}-\boldsymbol{y})] = \sum_{\ell=1}^{L} w_\ell \mathbb{E}(\widehat{y}_\ell - y_\ell)^2, \quad (30)$$

where $G := \text{diag}(w_1,\ldots,w_L)$ is the matrix assigning the link weights. After some algebra we get:

$$\begin{aligned}&\mathbb{E}[(\widehat{\boldsymbol{y}}-\boldsymbol{y})^T G(\widehat{\boldsymbol{y}}-\boldsymbol{y})]\\ &\quad=\mathbb{E}[(G^{1/2}\widehat{\boldsymbol{y}}-G^{1/2}\boldsymbol{y})^T(G^{1/2}\widehat{\boldsymbol{y}}-G^{1/2}\boldsymbol{y})]\\ &\quad=\mathbb{E}[(\widehat{\boldsymbol{y}}^G-\boldsymbol{y}^G)^T(\widehat{\boldsymbol{y}}^G-\boldsymbol{y}^G)],\end{aligned} \quad (31)$$

where $\widehat{\boldsymbol{y}}^G := G^{1/2}\widehat{\boldsymbol{y}}$ and $\boldsymbol{y}^G := G^{1/2}\boldsymbol{y}$. This shows that the trace-optimal solution $\widehat{\boldsymbol{y}}^G$ with the new "routing"



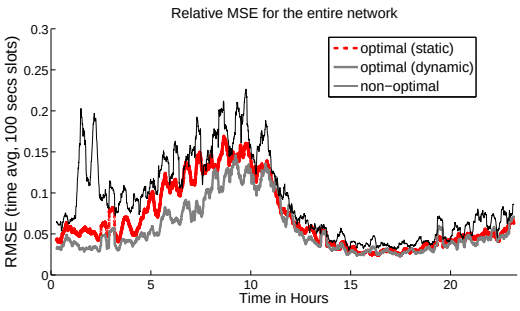


Fig. 9. Kriging on Internet2. (Data for 03/17/09.)

matrix $A^G := G^{1/2}A$ optimizes (31). We can apply our fast algorithms to approximate the standard trace-optimal solution $\widehat{\boldsymbol{y}}^G$. The optimal predictor for the *weighted* trace criterion in (30) is then obtained by $\widehat{\boldsymbol{y}} = G^{-1/2}\widehat{\boldsymbol{y}}^G$.

Fig. 6(b) shows the (weighted) prediction error (*i.e.*, a weighted ReMSE) for the scenario where the importance of monitoring links WASH $\rightarrow$ NEWY and WASH $\rightarrow$ ATLA elevates. We can model this by assigning unit weights to all other links, and a weight of ten to the important ones. Based on Eq. (28) and the spectrum shown in Fig. 6(a) we select a budget of $K = 7$. The solution given by PCAPH (which actually coincides with the exact optimal solution) includes the high importance links, plus links LOSA – HOUS (both directions), KANS – CHIC (both directions) and SALT $\rightarrow$ KANS. Fig. 6(b) clearly shows that the traffic prediction based on this set of links is way more accurate than the one based on a randomly chosen set (that includes the important links too).

VIII. Discussion

In conclusion, large-scale optimal monitoring is an important, yet computationally challenging problem, suitable for applications including anomaly detection in communication networks and environmental monitoring using sensor networks. We developed *fast algorithms* for approximate solutions that can be applied to large scale networks in real time. Randomized implementations of the algorithms in a distributed or parallel fashion can even yield the optimal solution in a fraction of the time needed to obtain exact solutions using integer programming techniques. Moreover, the novel PCA-based error lower bounds are practical, network-specific alternatives to existing theoretical lower bounds (see [15]), and help us assess the quality of approximation. We discuss how budget-selection can be achieved via PCA, and have applied our algorithms in numerous scenarios. The conclusion drawn is that our methods highly outperform traditional greedy techniques (see [4]), and hence are amenable and more scalable for online implementation in dynamical environments.

Appendix A
Principal Component Analysis theorems

Let $\Sigma_y = AA^T$ be the covariance matrix of the vector of all links $\boldsymbol{y}$. SVD of A yields $A = UD^{1/2}V^T$, with $D^{1/2} := \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$, $p = \min\{L, J\}$, where σ_i's are the singular values of A. Thus, $(\lambda_1 \geq \cdots \geq \lambda_p) \equiv (\sigma_1^2 \geq \cdots \geq \sigma_p^2)$ are the eigenvalues of the $J \times J$ matrix $\Omega \equiv A^TA$. Recall that the columns of A^T are denoted by $\boldsymbol{a}_\ell \in \mathbb{R}^J, \ell = 1, \cdots, L$.

Theorem 5 (Trace)**.** *Let P_W denote the projection matrix onto a sub-space $W \leq \mathbb{R}^J$. Then,*

$$\min_{W \leq \mathbb{R}^J,\ \dim(W)=K} \sum_{\ell=1}^{L} \|\boldsymbol{a}_\ell - P_W \boldsymbol{a}_\ell\|^2 = \sum_{j=K+1}^{p} \lambda_j,$$

where the lower bound is achieved for the sub-space $W^* = \mathrm{span}(\mathbf{v}_1, \cdots, \mathbf{v}_K)$ of the eigenvectors corresponding to the largest K eigenvalues of Ω.

The proof of the theorem is given in [9].

Proof of Theorem 1: We will use Theorem 5. Recall that $\Sigma_{\mathrm{err}}(\mathcal{O}) = A(I - P_\mathcal{O})A^T$, where $P_\mathcal{O} := A_o^T(A_oA_o^T)^{-1}A_o$. The projection matrix $(I - P_\mathcal{O})$ is symmetric and indempotent (*i.e.*, $(I-P_\mathcal{O}) = (I-P_\mathcal{O})^2$), so we have $\Sigma_{\mathrm{err}} = [(I - P_\mathcal{O})A^T]^T[(I - P_\mathcal{O})A^T]$, and therefore

$$\mathrm{trace}(\Sigma_{\mathrm{err}}(\mathcal{O})) = \sum_{\ell=1}^{L} \|(I-P_\mathcal{O})\boldsymbol{a}_\ell\|^2 = \sum_{\ell=1}^{L} \mathrm{dist}(\boldsymbol{a}_\ell, W_\mathcal{O})^2 \tag{32}$$

where $W_\mathcal{O} := \mathrm{span}(\boldsymbol{a}_\ell,\ \ell \in \mathcal{O})$ is the sub-space spanned by the vectors $\boldsymbol{a}_\ell$ corresponding to the observed links. The vector $(I - P_\mathcal{O})\boldsymbol{a}_\ell$ is the "perpendicular" dropped from point $\boldsymbol{a}_\ell$ to the hyperplane $\mathrm{Range}(A_o^T)$. Hence $\|(I - P_\mathcal{O})\boldsymbol{a}_\ell\|$ is the distance from $\boldsymbol{a}_\ell$ to $\mathrm{Range}(A_o^T) = \mathrm{span}(\boldsymbol{a}_\ell,\ \ell \in \mathcal{O})$, where $A_o^T = (\boldsymbol{a}_\ell)_{\ell \in \mathcal{O}}$.

Using Theorem 5 we see that the sum of (32) is minimized when the projection matrix $P_\mathcal{O}$ equals $P_K =$

$V_K V_K^T$. Hence,

$$\begin{aligned}\mathrm{trace}(\Sigma_{\mathrm{err}}(\mathcal{O})) &= \mathrm{trace}[A(I-P_{\mathcal{O}})A^T] \\ &\geq \|A - AP_K\|_F^2 = \sum_{i=K+1}^{p} \lambda_i.\end{aligned}$$

■

Similar result holds for the spectral norm case:

Theorem 6 (Spectral norm)**.**

$$\min_{\mathrm{rank}(B)=K} \|A-B\|_2 = \|A - AP_K\|_2 \qquad (33)$$

where P_K is the projection matrix $P_K = V_K V_K^T$. The columns of matrix V_K are the top K right singular vectors of A, i.e.[10]*, $P_K = V\mathrm{diag}(\mathbf{1}_K^T, \mathbf{0}_{L-K}^T)V^T$.*

For the proof of this result, see Theorem 2.5.3 in [27].

Proof of Theorem 2: Using Theorem 6, we need to show that:

$$\rho(\Sigma_{\mathrm{err}}(\mathcal{O})) = \rho[A(I-P_{\mathcal{O}})A^T] \geq \|A-AP_K\|_2^2 = \lambda_{K+1}.$$

We first calculate the lower bound when $P_K = V_K V_K^T$. We use the SVD of A, $A = UD^{1/2}V^T$ and the projection matrix $I - P_K = V\mathrm{diag}(\mathbf{0}_K^T, \mathbf{1}_{L-K}^T)V^T$. Thus, we have

$$\begin{aligned}A - AP_K &= UD^{1/2}V^T V\mathrm{diag}(\mathbf{0}_K^T, \mathbf{1}_{L-K}^T)V^T \\ &= UD^{1/2}\mathrm{diag}(\mathbf{0}_K^T, \mathbf{1}_{L-K}^T)V^T \\ &= U\mathrm{diag}(\mathbf{0}_K^T, \sigma_{K+1}, \ldots, \sigma_L)V^T.\end{aligned}$$

By the definition of spectral norm:

$$\begin{aligned}\|A - AP_K\|_2^2 &= \rho\left((U\mathrm{diag}(\mathbf{0}_K^T, \sigma_{K+1}, \ldots, \sigma_L)V^T)^T\right. \\ &\qquad \left.\times U\mathrm{diag}(\mathbf{0}_K^T, \sigma_{K+1}, \ldots, \sigma_L)V^T\right) \\ &= \rho\left(V\mathrm{diag}(\mathbf{0}_K^T, \lambda_{K+1}, \ldots, \lambda_L)V^T\right) \\ &= \lambda_{K+1}.\end{aligned}$$

Consequently, using also Theorem 6 we obtain:

$$\begin{aligned}\rho(\Sigma_{\mathrm{err}}(\mathcal{O})) &= \rho\left[A(I-P_{\mathcal{O}})A^T\right] \\ &= \rho\left[(A-AP_{\mathcal{O}})(A-AP_{\mathcal{O}})^T\right] \\ &= \|A - AP_{\mathcal{O}}\|_2^2 \\ &\geq \|A - AP_K\|_2^2 \\ &= \lambda_{K+1}.\end{aligned}$$

■

Appendix B
Proposition Proofs

Proof of Proposition 1: The proof resembles the Gram–Schmidt orthogonalization procedure. We will show by induction that

$$A^{(k)}A^{(k)^T} = A(I - A_o^T(A_o A_o^T)^{-1}A_o)A^T. \qquad (34)$$

[10]We symbolize the vector of ones of dimension k as $\mathbf{1}_k$ and the vector of zeros as $\mathbf{0}_k$.

For notational simplicity, let $\mathcal{O} := \mathcal{O}^k$, i.e., we drop the subscript k of the matrix A_{o_k}. It is easy to see that the sequential updates of the matrices $A^{(k)}$ in (13) can be represented in matrix form as follows:

$$A^{(k)} = AP_1P_2\cdots P_k, \qquad (35)$$

where

$$P_1 = I - \frac{\boldsymbol{a}_{i_1}\boldsymbol{a}_{i_1}^T}{\|\boldsymbol{a}_{i_1}\|^2},\ \cdots, P_k = I - \frac{\boldsymbol{a}_{i_k}^{(k-1)}\boldsymbol{a}_{i_k}^{(k-1)^T}}{\|\boldsymbol{a}_{i_k}^{(k-1)}\|^2}.$$

Here $\boldsymbol{a}_i^{(k)} \in \mathbb{R}^J$ denote the rows of the matrix $A^{(k)}$, where by convention $A^{(0)} = A$. Observe that P_k is the orthogonal projection matrix onto the space $(\mathrm{span}\{\boldsymbol{a}_{i_k}^{(k-1)}\})^{\perp}$, i.e. the orthogonal complement of the one-dimensional space spanned by $\boldsymbol{a}_{i_k}^{(k-1)}$. Here $\boldsymbol{a}_{i_k}^{(k-1)}$ corresponds to the link i_k added to the set $\mathcal{O}$ on step k.

This shows that on the $k-$th step *all* the rows of the matrix $A^{(k)}$ are *orthogonal* to $\mathrm{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\} \equiv \mathrm{span}\{\boldsymbol{a}_{i_1}^{(0)}, \cdots, \boldsymbol{a}_{i_k}^{(k-1)}\}$. The last subspace is generated by the vectors corresponding to the links $\{i_1, \cdots, i_k\}$ added on the first k steps.

Now, to complete the proof, it is enough to show that

$$P_1P_2\cdots P_k = P_{\mathrm{span}(\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k})^{\perp}}. \qquad (36)$$

Indeed, we have that

$$P_{(\mathrm{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\})^{\perp}} = I - A_o^T(A_o A_o^T)^{-1}A_o,$$

where $A_o^T = (\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k})$. Therefore, by (35) and (36), one obtains (34), which yields (14).

We now prove (36) by induction.

Induction Basis: Relation (36) trivially holds for $k = 1$.

Induction Hypothesis: Suppose that (36) holds.

Induction Step: We will show that (36) holds with k replaced by $k+1$.

Note that by the induction hypothesis $\boldsymbol{a}_{i_{k+1}}^{(k)}$ is the orthogonal projection of $\boldsymbol{a}_{i_{k+1}}$ onto $(\mathrm{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\})^{\perp}$. Therefore,

$$\begin{aligned}&\mathrm{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}, \boldsymbol{a}_{i_{k+1}}\} \\ &\qquad = \mathrm{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\} \oplus \mathrm{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\},\end{aligned}$$

where $\oplus$ denotes sum of orthogonal subspaces of $\mathbb{R}^J$. This shows that

$$P_{\mathrm{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}, \boldsymbol{a}_{i_{k+1}}\}} = P_{\mathrm{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\}} + P_{\mathrm{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}}.$$

Since $P_{W^{\perp}} = I - P_W$, we obtain

$$\begin{aligned}&P_{(\mathrm{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}, \boldsymbol{a}_{i_{k+1}}\})^{\perp}} \qquad (37) \\ &\qquad = I - P_{\mathrm{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\}} - P_{\mathrm{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}}.\end{aligned}$$

Note, however, that since $\boldsymbol{a}_{i_{k+1}}^{(k)} \perp \text{span}\{\boldsymbol{a}_{i_1}, \cdots, \boldsymbol{a}_{i_k}\}$, we have $P_{\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k}\}} P_{\text{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}} = 0$, and the right-hand side of (37) equals

$$(I - P_{\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k}\}})(I - P_{\text{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}}) = P_{\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k}\}^\perp} P_{\text{span}\{\boldsymbol{a}_{i_{k+1}}^{(k)}\}^\perp}.$$

This, in view of the induction hypothesis and (37), implies that

$$P_{(\text{span}\{\boldsymbol{a}_{i_1},\cdots,\boldsymbol{a}_{i_k},\boldsymbol{a}_{i_{k+1}}\})^\perp} = P_1 \cdots P_k P_{k+1},$$

which completes the proof of the induction step. ∎

Proof of Proposition 2: By definition, $\tilde{\lambda}_1$ is the following random variable:

$$\tilde{\lambda}_1 := \max\{\boldsymbol{z}_1^T \Sigma \boldsymbol{z}_1, ..., \boldsymbol{z}_m^T \Sigma \boldsymbol{z}_m\}, \tag{38}$$

where $\boldsymbol{z}_i \in \mathbf{R}^L$, $i = 1, ..., m$ are *iid* random variables distributed on the unit sphere $\mathbf{S}^{L-1}$. Let the SVD of Σ be $\Sigma = VDV^T$, with the columns of V being the singular vectors of Σ and $D = \text{diag}(\lambda_1, ..., \lambda_\text{L})$. We use the fact that for every orthogonal matrix $P_{L\times L}$ the rotated vectors $\boldsymbol{w}_i = P\boldsymbol{z}_i$, $i = 1, ..., m$ are also *iid* random variables on the unit sphere $\mathbf{S}^{L-1}$. Then, for $P = V^T$,

$$\begin{aligned}\tilde{\lambda}_1 &= \max_i\{\boldsymbol{w}_i^T P\Sigma P^T \boldsymbol{w}_i\} = \max_i\{\boldsymbol{w}_i^T D \boldsymbol{w}_i\} \\ &= \max_i\{\lambda_1 w_i^2(1) + \lambda_2 w_i^2(2) + \ldots + \lambda_L w_i^2(L)\} \\ &\overset{w.p.1}{\geq} \max_i\{\lambda_1 w_i^2(1)\},\end{aligned} \tag{39}$$

where $w_i(k)$ is the k-th component of the vector $\boldsymbol{w}_i$. Then,

$$\begin{aligned}\mathbf{P}(\tilde{\lambda}_1 > \lambda_1(1-\varepsilon)) &\geq \mathbf{P}(\max_i\{\lambda_1 w_i^2(1)\} > \lambda_1(1-\varepsilon)) \\ &= 1 - \mathbf{P}(\max_i\{w_i^2(1)\} \leq (1-\varepsilon)).\end{aligned} \tag{40}$$

A moment's reflection shows that the distribution of the r.v. $w_i^2(1), \forall i$ is identical with the distribution of $z_i^2(1), \forall i$. By construction, for all i

$$z_i^2(1) = \frac{x_i^2(1)}{x_i^2(1) + x_i^2(2) + \ldots + x_i^2(L)} \tag{41}$$

where $x_i(j)$, $j = 1, ..., L$, $i = 1, ..., m$ are *iid* r.v. from the normal distribution $\mathcal{N}(0, 1)$.

It is well known (see [31], Section II.3) that if $X_1, X_2, \ldots, X_n$ are mutually independent Gaussian r.v. with expectation 0 and variance σ^2, then $X_1^2 + X_2^2 + \ldots + X_n^2$ follows the *Gamma* distribution with parameter $\alpha = 1/(2\sigma^2)$ and $\nu = n/2$. Therefore, for all $i = 1, \ldots, m$

$$x_i^2(1) \sim \text{Gamma}(\frac{1}{2}, \frac{1}{2}) \tag{42}$$

$$x_i^2(2) + \ldots + x_i^2(L) \sim \text{Gamma}(\frac{1}{2}, \frac{\text{L}-1}{2}). \tag{43}$$

It is also known that if X, Y are random variables with distribution Gamma(α, ν) and Gamma(β, ν), respectively, then the r.v. $X/(X+Y)$ follows the *Beta* distribution $\text{Beta}(\alpha, \beta)$. Hence, for all $i = 1, ..., m$

$$\frac{x_i^2(1)}{x_i^2(1) + x_i^2(2) + \ldots + x_i^2(L)} \sim \text{Beta}(\frac{1}{2}, \frac{\text{L}-1}{2}). \tag{44}$$

Continuing from Eq. (40)

$$\begin{aligned}&\mathbf{P}(\max_i\{w_i^2(1)\} \leq 1-\varepsilon) = [\mathbf{P}(w_1^2(1) \leq 1-\varepsilon)]^m \\ &= \Big(\int_0^{1-\varepsilon} f(x)dx\Big)^m = \Big(1 - \int_{1-\varepsilon}^1 f(x)dx\Big)^m \\ &\leq e^{-m(\int_{1-\varepsilon}^1 f(x)dx)},\end{aligned} \tag{45}$$

where $f(x)$ is the density function of the $\text{Beta}(1/2, (\text{L}-1)/2)$ distribution, and in the last inequality we used the fact that $1 - x \leq e^{-x}$, $\forall x \in \mathbf{R}$.

We now bound the integral in the exponent.

$$\begin{aligned}&\int_{1-\varepsilon}^1 \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}dx \\ &= \frac{\Gamma(L/2)}{\Gamma(1/2)\Gamma((L-1)/2)} \int_{1-\varepsilon}^1 x^{-\frac{1}{2}}(1-x)^{\frac{L-3}{2}}dx \\ &\leq \frac{\Gamma(L/2)}{\sqrt{\pi}\Gamma(-1/2+L/2)}(1-\varepsilon)^{-\frac{1}{2}} \int_{1-\varepsilon}^1 (1-x)^{\frac{L-3}{2}}dx.\end{aligned} \tag{46}$$

After calculating the integral and some algebra, the result follows. ∎

Proof of Proposition 3: By definition, $\tilde{\lambda}_1$ is the following random variable:

$$\tilde{\lambda}_1 := \max\{\frac{\boldsymbol{x}_1^T}{\|\boldsymbol{x}_1\|}\Sigma\frac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|}, \ldots, \frac{\boldsymbol{x}_m^T}{\|\boldsymbol{x}_m\|}\Sigma\frac{\boldsymbol{x}_m}{\|\boldsymbol{x}_m\|}\}, \tag{47}$$

with $\boldsymbol{x}_i \in \mathbf{R}^L$, $i = 1, ..., m$ are independent random vectors from the multivariate normal distribution $\mathcal{N}(0, I_L)$ Let the SVD of Σ be $\Sigma = VDV^T$, with the columns of V being the singular vectors of Σ and $D = \text{diag}(\lambda_1, ..., \lambda_\text{L})$. We multiply each vector $\boldsymbol{x}_i$ with the orthogonal matrix P, with $P = V^T$ to get $\boldsymbol{w}_i = P\boldsymbol{x}_i$. Since this operation preserves inner products, angles and distances, the vectors $\boldsymbol{w}_i$ are also *iid* from the $\mathcal{N}(0, I_L)$ distribution. Hence,

$$\begin{aligned}\tilde{\lambda}_1 &= \max_i\{\frac{\boldsymbol{x}_i^T}{\|\boldsymbol{x}_i\|}VDV^T\frac{\boldsymbol{x}_i}{\|\boldsymbol{x}_i\|}\} = \max_i\{\frac{\boldsymbol{w}_i^T}{\|\boldsymbol{w}_i\|}D\frac{\boldsymbol{w}_i}{\|\boldsymbol{w}_i\|}\} \\ &= \max_i\{\frac{\sum_{j=1}^L \lambda_j w_i^2(j)}{\|\boldsymbol{w}_i\|^2}\} = \lambda_1 \max_i\{\frac{\sum_{j=1}^L c_j w_i^2(j)}{\|\boldsymbol{w}_i\|^2}\}.\end{aligned} \tag{48}$$

Given $\varepsilon > 0$, the precision of approximation is:

$$\mathbf{P}(\tilde{\lambda}_1 > \lambda_1(1-\varepsilon)) = \mathbf{P}(\max_i \xi_i > 1-\varepsilon), \tag{49}$$

with the random variables $\xi_i = \sum_{j=1}^L c_j w_i^2(j)/\|\boldsymbol{w}_i\|^2$, $i = 1, \ldots, m$ being independent and identically distributed.

[16] Proceeding, we have

$$\mathbf{P}(\max_i \xi_i > 1-\varepsilon) = 1 - \mathbf{P}(\max_i \xi_i \le 1-\varepsilon) = 1 - [\mathbf{P}(\xi_1 \le 1-\varepsilon)]^m, \quad (50)$$

with ξ_1 chosen without loss of generality. Explicitly writing ξ_1 we obtain,

$$\begin{aligned}
\mathbf{P}(\xi_1 \le 1-\varepsilon) &= \mathbf{P}\Big(\sum_{j=1}^{L} c_j w_1^2(j) \le (1-\varepsilon)\sum_{j=1}^{L} w_1^2(j)\Big) \\
&= \mathbf{P}\Big(\sum_{j=1}^{t}[c_j-(1-\varepsilon)]w_1^2(j) \le \sum_{j=t+1}^{L}[(1-\varepsilon)-c_j]w_1^2(j)\Big) \\
&\le \mathbf{P}\Big(\sum_{j=1}^{t}[c_j-(1-\varepsilon)]w_1^2(j) \le \sum_{j=t+1}^{L}(1-\varepsilon)w_1^2(j)\Big) \\
&\le \mathbf{P}\Big([c_t-(1-\varepsilon)]\sum_{j=1}^{t} w_1^2(j) \le (1-\varepsilon)\sum_{j=t+1}^{L} w_1^2(j)\Big), \quad (51)
\end{aligned}$$

where we used the fact that $c_i > 1-\varepsilon$ for all $i \le t$, and $c_i \le 1-\varepsilon$ otherwise, to obtain the last two inequalities.

The random variables $\sum_{j=1}^{t} w_1^2(j)$ have *chi-square* density with t-degrees of freedom (see [31], Section II.3). Similarly, he random variables $\sum_{j=t+1}^{L} w_1^2(j)$ have *chi-square* density with $(L-t)$-degrees of freedom. Using the fact (see [31], Section II.3) that the random variable $F = (X/\nu_1)/(Y/\nu_2)$ – with X, Y being *chi-squared* distributed with ν_1 and ν_2 degrees of freedom respectively – has the F-density with parameters ν_1 and ν_2, the result follows.

■

APPENDIX C
ERROR REDUCTION BOUNDS

Proof of Theorem 3: For ease of notation, we drop the superscript k. For the same reason, we introduce the operator $\langle \boldsymbol{e}_1, \boldsymbol{e}_2 \rangle$ to represent the inner product between two vectors. Let $x_j := |\langle \boldsymbol{a}_j, \boldsymbol{a}_i \rangle|/\|\boldsymbol{a}_i\|$ be the length of the projection of any vector $\boldsymbol{a}_j$ to the selected vector $\boldsymbol{a}_i$. Also, let $y_j := |\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)} \rangle|$ be the projection length on $\mathbf{v}_1^{(k)}$.

Note that $\boldsymbol{a}_i/\|\boldsymbol{a}_i\| = \langle \boldsymbol{a}_i/\|\boldsymbol{a}_i\|, \mathbf{v}_1^{(k)} \rangle \mathbf{v}_1^{(k)} + \boldsymbol{a}_i^{\perp} = \gamma \mathbf{v}_1^{(k)} + \boldsymbol{a}_i^{\perp}$. Proceeding, we have

$$\begin{aligned}
x_j^2 &= |\langle \boldsymbol{a}_j, \gamma \mathbf{v}_1^{(k)} + \boldsymbol{a}_i^{\perp} \rangle|^2 \quad (52)\\
&= |\gamma \langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)} \rangle + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle|^2 \\
&\ge |\gamma |\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)} \rangle| - |\langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle||^2 \\
&\ge \gamma^2 \langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)} \rangle^2 - 2\gamma |\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)} \rangle||\langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle| \\
&= \gamma^2 y_j^2 - 2\gamma y_j |\langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle|,
\end{aligned}$$

after substitution of the projection length y_j on the first principal axis. Observe that due to orthogonality, we have $\gamma^2 + \|\boldsymbol{a}_i^{\perp}\|^2 = 1$. Using the Cauchy – Bunyakovsky – Schwarz inequality we then bound the term $|\langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle|$ as $|\langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle| \le \|\boldsymbol{a}_j\| \|\boldsymbol{a}_i^{\perp}\| = \|\boldsymbol{a}_j\| \sqrt{1-\gamma^2}$. Therefore,

$$x_j^2 \ge \gamma^2 y_j^2 - 2\gamma y_j \|\boldsymbol{a}_j\| \sqrt{1-\gamma^2}. \quad (53)$$

Now we sum over all $j \in \mathcal{L}$ to get the error reduction for all links. We thus get,

$$\begin{aligned}
\sum_{j=1}^{L} x_j^2 &\ge \gamma^2 \sum_{j=1}^{L} y_j^2 - 2\gamma\sqrt{1-\gamma^2} \sum_{j=1}^{L} y_j \|\boldsymbol{a}_j\| \quad (54)\\
&= \gamma^2 \lambda_1^{(k)} - 2\gamma\sqrt{1-\gamma^2} \sum_{j=1}^{L} y_j \|\boldsymbol{a}_j\|,
\end{aligned}$$

where in the last equation we used Theorem 1 for $K = 1$ on matrix $A^{(k)}$; thus, the PCA error reduction equals the largest eigenvalue, $\lambda_1^{(k)}$.

We now isolate the term $\sum_{j=1}^{L} y_j \|\boldsymbol{a}_j\|$ and using again the Cauchy – Bunyakovsky – Schwarz bound we get

$$\begin{aligned}
\sum_{j=1}^{L} y_j \|\boldsymbol{a}_j\| &\le \sqrt{\sum_{j=1}^{L} (y_j)^2} \sqrt{\sum_{j=1}^{L} \|\boldsymbol{a}_j\|^2} \quad (55)\\
&= \sqrt{\lambda_1^{(k)}} \sqrt{\mathrm{trace}(A^{(k)^T} A^{(k)})} \\
&= \sqrt{\lambda_1^{(k)}} \sqrt{\sum_{j=1}^{L} \lambda_j^{(k)}}.
\end{aligned}$$

The result follows. ■

Proof of Theorem 4: Again, when there is no ambiguity we drop the superscript k. Let $x_j := |\langle \boldsymbol{a}_j, \boldsymbol{a}_i \rangle|/\|\boldsymbol{a}_i\|$ be the length of the projection of any vector $\boldsymbol{a}_j$ to the selected by the algorithm vector $\boldsymbol{a}_i$. Also, let $y_j := |\langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)} \rangle|$ be the projection length on $\mathbf{v}_1^{(k)}$.

As before $\boldsymbol{a}_i/\|\boldsymbol{a}_i\| = \gamma \mathbf{v}_1^{(k)} + \boldsymbol{a}_i^{\perp}$. Proceeding, we have

$$\begin{aligned}
x_j^2 &= |\langle \boldsymbol{a}_j, \gamma \mathbf{v}_1^{(k)} + \boldsymbol{a}_i^{\perp} \rangle|^2 \quad (56)\\
&= |\gamma \langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)} \rangle + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle|^2 \\
&= \gamma^2 \langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)} \rangle^2 + 2\gamma \langle \boldsymbol{a}_j, \mathbf{v}_1^{(k)} \rangle \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle^2 \\
&= \gamma^2 y_j^2 + 2\gamma y_j \underbrace{\langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle} + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle^2 \\
&\le \gamma^2 y_j^2 + 2\gamma \underbrace{y_j \|\boldsymbol{a}_j\|} \sqrt{1-\gamma^2} + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle^2 \\
&\le \gamma^2 y_j^2 + 2\gamma \sqrt{\lambda_1^{(k)}} \sqrt{\sum_{j=1}^{L} \lambda_j^{(k)}} \sqrt{1-\gamma^2} + \langle \boldsymbol{a}_j, \boldsymbol{a}_i^{\perp} \rangle^2, \quad (57)
\end{aligned}$$

with the last two lines obtained using the Cauchy – Bunyakovsky – Schwarz inequality on the specified terms.

Note that for $\phi > 0$, $\boldsymbol{a}_j = \phi \mathbf{v}_1^{(k)} + \boldsymbol{a}_j^{\perp}$, i.e. $\boldsymbol{a}_j^{\perp} \perp \mathbf{v}_1^{(k)}$. Then, $\langle \phi \mathbf{v}_1^{(k)} + \boldsymbol{a}_j^{\perp}, \boldsymbol{a}_i^{\perp} \rangle = \langle \boldsymbol{a}_j^{\perp}, \boldsymbol{a}_i^{\perp} \rangle$ because $\boldsymbol{a}_i^{\perp} \perp \mathbf{v}_1^{(k)}$ as well. Moreover, notice that $|\langle \boldsymbol{a}_j^{\perp}, \boldsymbol{a}_i^{\perp} \rangle| \leq |\langle \boldsymbol{a}_j^{\perp}, \boldsymbol{a}_i^{\perp} \rangle| / \|\boldsymbol{a}_i^{\perp}\|$ because $\|\boldsymbol{a}_i^{\perp}\| = \sqrt{1-\gamma^2} \leq 1$ (by construction of $\|\boldsymbol{a}_i^{\perp}\|$). These steps suggest that:

$$\sum_{j=1}^{L} \langle \boldsymbol{a}_j^{\perp}, \frac{\boldsymbol{a}_i^{\perp}}{\|\boldsymbol{a}_i^{\perp}\|} \rangle^2 \leq \sum_{j=1}^{L} \langle \boldsymbol{a}_j^{\perp}, \mathbf{v}_2^{(k)} \rangle^2 = \lambda_2^{(k)}. \qquad (58)$$

This result follows from PCA. It basically says that the sum of the squares of the projection lengths of any vector $\boldsymbol{a}_j^{\perp}$ on $\boldsymbol{a}_i^{\perp}$ is bounded by the sum of the squares of the projections on the second principal axis, $\mathbf{v}_2^{(k)}$. In other words, $\mathbf{v}_2^{(k)}$ gives the maximum projection with respect to any other vector from $\mathrm{Null}(\mathbf{v}_1^{(k)^T})$. Summing both sides of (57) over all $j = 1, ..., L$, and using (58), concludes the proof.

■

APPENDIX D
NP-HARDNESS

We prove NP-hardness using reduction from the NP-hard problem *Subset Selection for Regression* [17].

Definition 1 (Subset Selection for Regresssion)**.** *Given matrix C, vector b and k, select a set $\mathcal{S}$ of k observation variables that minimize the mean square prediction error* $\mathrm{Err}(Z, S) = \mathrm{Var}(Z) - \boldsymbol{b}_S^T C_S^{-1} \boldsymbol{b}_S$.

Z is the predictor variable, and $X_1, ..., X_n$ are the observation variables. The covariance matrix of the observation variables is denoted by C. Vector $\boldsymbol{b}$ denotes the covariances between Z and X_i, $i = 1, ..., n$.

Proof of NP-hardness:

Clearly, the decision version of our problem is in NP. This is because given a *candidate* solution $\mathcal{O}$, we can decide in polynomial time whether or not the prediction error using the given set $\mathcal{O}$ is greater or less than a given value. We will use the method of *Restriction* [14] to show that every instance of the problem in [17] is a special case of an analogous to (6) problem.

Consider the problem of selecting a subset $\mathcal{O}$ of links in order to best predict the traffic at a *specific* link i, $i \notin \mathcal{O}$. W.l.o.g, say that $i = 1$, i.e. we want to best predict traffic at link 1. Then, the optimization problem is to select $\mathcal{O} \subset \mathcal{L} \setminus \{1\}$ that minimizes the following:

$$\mathrm{trace}(\Sigma_{11} - \Sigma_{1,\mathrm{o}} \Sigma_{\mathrm{o,o}}^{-1} \Sigma_{\mathrm{o},1}) \text{ or} \qquad (59)$$

$$\rho(\Sigma_{11} - \Sigma_{1,o} \Sigma_{o,o}^{-1} \Sigma_{o,1}) \qquad (60)$$

which correspond to our A- and E-optimality criteria, respectively. But the quanity in parenthesis is a scalar, so the trace and spectral norm operations are redundant. Going back to the *subset selection for regression* we observe the following: Variable Z corresponds to traffic at link 1, i.e. it corresponds to variable y_1. The vector $\boldsymbol{b}$ corresponds to the first column (or row) of matrix Σ, i.e. $\Sigma_{:,1}$. Matrix C corresponds to $\Sigma_{\hat{L}}$, where $\hat{\mathcal{L}} = \mathcal{L} \setminus \{1\}$. Hence, using the notation of our problem, the problem in [17] aims to minimize, by choosing a set $\mathcal{O}$ with $|\mathcal{O}| = k$, the error $\Sigma_{11} - \Sigma_{1,o} \Sigma_{o,o}^{-1} \Sigma_{o,1}$. Thus, every instance of the NP-hard problem in [17] is a special case of our problem. Because our problem is in NP, it is then an NP-hard problem.

■

REFERENCES

[1] Cisco Systems, "Cisco IOS Netflow," www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html.

[2] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 219–230, August 2004.

[3] C. Boutsidis, M. W. Mahoney, and P. Drineas, "Unsupervised feature selection for principal components analysis," in *Proceeding of the 14th ACM SIGKDD*, 2008, pp. 61–69.

[4] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, June 2008.

[5] T. Gabriel and Crainic, "Service network design in freight transportation," *European Journal of Operational Research*, vol. 122, no. 2, pp. 272 – 288, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221799002337

[6] I. C. Paschalidis and G. Smaragdakis, "Spatio-temporal network anomaly detection by assessing deviations of empirical measures," *IEEE/ACM Trans. Netw.*, vol. 17, pp. 685–697, June 2009.

[7] H. Singhal and G. Michailidis, "Optimal sampling in state space models with applications to network monitoring," in *Proceedings of the 2008 ACM SIGMETRICS*, 2008, pp. 145–156.

[8] S. Stoev, G. Michailidis, and J. Vaughan, "On global modeling of backbone network traffic," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –5.

[9] ——, "On global modeling of network traffic," University of Michigan, Tech. Rep., 2010, http://www.stat.lsa.umich.edu/∼sstoev/global_tr.pdf.

[10] D. Chua, E. Kolaczyk, and M. Crovella, "Network kriging," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 12, pp. 2263 –2272, dec. 2006.

[11] Internet2, "`http://www.internet2.edu/observatory/`."

[12] C.-W. Ko, J. Lee, and M. Queyranne, "An Exact Algorithm for Maximum Entropy Sampling," *OPERATIONS RESEARCH*, vol. 43, no. 4, pp. 684–691, Jul. 1995. [Online]. Available: http://dx.doi.org/10.1287/opre.43.4.684

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, United Kingdom: Cambridge University Press, 2004.

[14] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[15] G. Nemhauser and L. Wolsey, "Maximizing submodular set functions: Formulations and analysis of algorithms," in *Annals of Discrete Mathematics*, 1981, vol. 59, pp. 279 – 301.

[16] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Mathematical Programming*, vol. 14, pp. 265–294, 1978.

[17] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *Proceedings of the 40th annual ACM symposium on Theory of computing*, 2008, pp. 45–54.

[18] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397 –3415, dec 1993.

[19] A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss, "Approximation of functions over redundant dictionaries using coherence," in *Proc. of 14th annual ACM-SIAM symp. on Discrete algorithms*, ser. SODA '03, 2003, pp. 243–252.

[20] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation: part i: Greedy pursuit," *Signal Process.*, vol. 86, pp. 572–588, March 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1140723.1140735

[21] Y. C. Pati, R. Rezaiifar, Y. C. P. R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, 1993, pp. 40–44.

[22] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, March 2003.

[23] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[24] D. Bajovic, B. Sinopoli, and J. Xavier, "Sensor selection for event detection in wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 59, no. 10, pp. 4938 –4953, oct. 2011.

[25] E. Tsakonas, J. Jalden, and B. Ottersten, "Semidefinite relaxations of robust binary least squares under ellipsoidal uncertainty sets," *Signal Processing, IEEE Transactions on*, vol. 59, no. 11, pp. 5169 –5180, nov. 2011.

[26] K. Park and W. Willinger, Eds., *Self-Similar Network Traffic and Performance Evaluation*. New York: J. Wiley & Sons, Inc., 2000.

[27] G. H. Golub and C. F. van Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, Oct. 1996.

[28] I. Jolliffe, *Principal Component Analysis (2nd ed.)*. New York: Springer-Verlag, 2002.

[29] A. L. Barabasi and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[30] J. Winick and S. Jamin, "Inet-3.0: Internet topology generator," University of Michigan, Tech. Rep., 2002, http://topology.eecs.umich.edu/inet/.

[31] W. Feller, *An Introduction to Probability Theory and its Applications (2nd ed.)*. New York: John Wiley and Sons, 1971.